ISSN 0716 4777

MICROBYTE

Vol IV Nº 2 TODO COMPUTACION Y TELECOMUNICACIONES JULIO 1987 Nº 35 $ 300



Programación estructurada

Archivos dBase a Word Perfect

Teoría de colas con Lotus 1, 2, 3

Computer Club

Chile
Huelén 164 B
Providencia
2231530 2239097
Télex 346304 MBYTE CK
Coordinador General
José Kaffman T.
Director Publicidad y RR PP
Ariel Leporati P.
Ventas
Orlando Zepeda
Diagramación y Producción Gráfica
Tintazul Publicidad
Directora de Arte
Paz Barba
Montaje
Pedro Arce
Germán Carvajal
Cuerpo Editorial
Jaime Aravena
Guillermo Beuchat
Carlos Contreras
Héctor Miranda
Humberto Silun
Corresponsales en el exterior
Luis Kaffman T. (Londres)
Alfredo Zerowsky (París)
Víctor Kahan (Ohio)
Fotocomposición
LASER
Distribución
Antártica S.A.
Impresión
Eme Cuatro

Argentina
Viamonte 723, Of. 2
3929460
1053 Cap. Federal
Télex 25390 VIDAL AR
Administración General
Judith Kaffman T.
Redacción y Crónica
Guillermo Javier Defranco
Representante Legal
Dr. Alfredo P. Carlomagno
Distribución
Distribuidor en Capital: TRI-BI-FER
San Nicolás 3169, Capital
Distribuidor interior: DGP
Hipólito Irigoyen 1450, Capital

Microbyte es una publicación mensual de KVC Asociados.

MICROBYTE

Foto Portada: La computación hacia lo desconocido

INDICE

LECTURAS

Título : PROLOG: Programación y aplicaciones en Inteligencia Artificial
Autor : A. A. Berk
Editorial: Anaya Multimedia (1986, 204 págs.)
Precio : $ 3.860

El PROLOG juega un papel importantísimo dentro de la Inteligencia Artificial, ya que será el Lenguaje nativo de las máquinas de quinta generación. La rápida aparición de dialectos de PROLOG para computadores personales (Micro Prolog, Turbo Prolog) hace posible para cualquier aficionado la experimentación con este potente lenguaje de tratamiento de listas en el ámbito de la I.A. debido a que el PROLOG es el lenguaje ideal para desarrollar sistemas expertos compactos y sofisticados.

PROLOG, Programación y aplicaciones en I.A. desarrolla y explica los fundamentos técnicos y estructuras de programación en PROLOG utilizando la implementación Micro PROLOG para explicar paso a paso cómo analizar y comprender el lenguaje natural, cómo crear bases de conocimiento y cómo lograr que el programa haga uso inteligente y creativo de la información allí almacenada.

A lo largo del libro se analizan todos los aspectos importantes del PROLOG necesarios para comenzar a desarrollar aplicaciones: Aritmética y procesamiento de listas, recursión, análisis sintáctico, lenguaje natural y programación metalógica. Le explicará con claridad cómo construir sofisticadas bases de conocimiento y le permitirá adentrarse ágilmente en el retador mundo de la Inteligencia Artificial.

Título : Lógica, Programación e Inteligencia Artificial
Autor : Robert Kowalski
Editorial: Díaz de Santos (1986, 412 págs.)
Precio : $ 8.568

El presente libro del profesor Kowalski constituye una de las obras fundamentales que debe tener todo profesional, investigador o estudiante de Inteligencia Artificial, tanto por la información que proporciona como por el nivel académico que encierra.

También será de gran utilidad para los Lógicos y Matemáticos que estén interesados en conocer cómo la Lógica es el pilar más importante de los lenguajes de nueva generación de los cuales el PROLOG es ya una realidad.

El objetivo de este libro es investigar la aplicación de la Lógica en resolución de problemas y a la programación de ordenadores. Para su estudio no se requiere un conocimiento previo de estos campos y por consiguiente representa una introducción muy adecuada a los temas de:

* Lógica
* Teoría de resolución de problemas y
* Programación de ordenadores.

El libro está escrito sin prestar una atención especial al aspecto formal y por ello se han omitido prácticamente las demostraciones. Para seguirlo no se necesita un conocimiento previo de lógica, resolución de problemas o ciencia de la computación por lo que es muy adecuado para estudiantes. Sin embargo muchos de los ejercicios tienen un nivel que puede ser considerado como avanzado.

Título : Comunicaciones y Redes de Procesamiento de Datos
Autor : Néstor González S.
Editorial: Mc Graw Hill (1987, 396 págs.)
Precio : $ 4.780

En el Prólogo del libro el autor agradece el tiempo dedicado por autoridades de ECOM CHILE a través del señor Eugenio Bonnefont y los doctores Jorge Olivos y Patricio Poblete, de la Universidad de Chile quienes colaboraron con su tiempo para optimizar el material ahora publicado.

Después de presentar una introducción y conceptos básicos el autor relaciona los elementos de un sistema de comunicaciones: adaptadores, modems, puentes, controladores, etc. y los medios físicos de transmisión, así como la detección y corrección de errores de transmisión. Las disciplinas de telecomunicaciones se ejemplifican tomando sólo algunas de la enorme variedad existente para ingresar al análisis de los Conceptos de redes de procesamientos de datos, redes locales LAN y redes públicas de datos PDN (Public Data Network) tomando como ejemplo la Red Pública de Datos de ECOM.

Los temas de SNA (System Network Architecture), DCNA (Data Communications Network Architecture) y DNA (Digital Network Architecture) son tratados en base a información de diferentes compañías proveedoras de equipos para finalizar con un capítulo dedicado a las redes privadas y servicios especiales.

Los libros comentados en esta sección pueden ser adquiridos en Galileo Libros
Dr. Barros Borgoño 9 - A
Teléfono 2238314 Providencia
Santiago - Chile

EDITORIAL

Que la computación es una herramienta que se ha introducido en todas las áreas de actividad, es una frase que por repetirse tanto ha caído en el campo de los cliches. Sin embargo, a pesar de esto, no pasa un día en que dejemos de asombrarnos de los nuevos campos que va invadiendo.

La ultima sorpresa con que me encontré y que quiero compartir, es haber conocido más de cerca los verdaderos milagros que se están haciendo en el terreno de la computación gráfica y la sorprendente alianza que ha desarrollado ésta con la inteligencia artificial para automatizar a niveles increibles la producción de imágenes animadas.

Los principales beneficiarios de una verdadera avalancha de técnicas y productos que están apareciendo en este campo son ingenieros, artistas y profesionales, quienes en sus estaciones de trabajo y computadores personales ven facilitada al máximo su labor de creación gráfica.

El mercado para el video es también inconmensurable. Lo que hoy se utiliza para la produccion de peliculas promete convertirse en un breve plazo en un valioso aporte para actividades tan distintas como la medicina, el desarrollo y protección de recursos energéticos, la educación, etc.

En efecto, la computación gráfica ha alcanzado un nuevo estadio, en el cual, gracias a la incorporación de técnicas derivadas de investigaciones en inteligencia artificial, es posible crear una animación completa a partir de una descripción escrita en un lenguaje cotidiano.

El usuario de estos sistemas va creando figuras y movimientos básicos, todo traducido a ecuaciones que van creando una biblioteca de conceptos. Para hacer correr a una figura no es necesario ya dibujar cada uno de los pasos. Es el computador el que se encarga de eso, para lo cual va tomando en cuenta además conceptos físicos tales como la ley de gravedad.

Uno de los desarrollos más interesantes en esta área lo constituye un paquete desarrollado por NEC en Japón, el cual generó por sí solo una película animada de la fábula de la tortuga y la liebre a partir de una simple descripción de la fábula escrita en japonés.

Uno de los aspectos en que es fácil medir el avance en las técnicas de animación es en su calidad. En general, la resolución o calidad de un cuadro de video desarrollado por computador se mide en polígonos. La película TRON, exhibida en 1983, utilizando los más modernos recursos de la época, contaba con unos 50 000 polígonos por cuadro. Las animaciones en la Guerra de las Galaxias ya contaban con unos 800 000 polígonos, mientras que actualmente las técnicas de animación ya permiten crear cuadros con 3 millones de polígonos, de una nitidez perfecta.

Si bien para esto se requieren aun de costosos equipos, la tendencia indica que dentro de pocos años ya podremos contar con sistemas semiprofesionales al alcance del hobbysta. Hoy ya existen sistemas desarrollados para modestos PC s que permiten desarrollar animaciones de buen nivel, entre ellos ArtWork, VideoWork y Animate.

NOTICIAS
NOVEDADES

Amstrad primero en ventas en Europe

Según estadísticas de International Data Corp Amstrad se convirtió en el principal vendedor de microcomputadores en Europa durante 1986 La mayor parte de las ventas apuntó al mercado hogareño estando IBM en el primer lugar en el segmento equipos profesionales

Las ventas de Amstrad aumentaron en un 127% en relación al año anterior De sus modelos PC Compatibles Amstrad logró vender 160 000 en tan sólo los ultimos cuatro meses de 1986 desde su lanzamiento en septiembre

Los exitos de Amstrad en sus tierras lo han hecho lanzarse a la conquista del mercado norteamericano aunque consciente del fracaso sufrido anteriormente por otros fabricantes británicos como Acorn o ACT lo está haciendo con bastante precaución

Los equipos lanzados por Amstrad al mercado norteamericano durante la exposición de Comdex en Atlanta son del nuevo modelo PC1640 el que difiere del PC1512 original en una mayor capacidad gráfica y tienen un valor que fluctua entre U$ 900 y U$ 2 000 de acuerdo a la configuración



Liquidación de software

Hasta ahora cuando se hablaba de software economico el nombre que surgía por excelencia era el de Borland quien se ha hecho famoso con sus excelentes compiladores Turbo-Pascal Turbo-Prolog o Turbo-C todos a U$ 99 95 cada uno

Ahora sin embargo una firma británica ha llegado a doblarle la mano y su oferta es sorprendente En efecto Psion Limited, conocida anteriormente por un computador de bolsillo el Organizer que traía incorporado su software y por haber desarrollado también el software básico del desafortunado QL de Sinclair, ha sacado al mercado PC-Four un set de cuatro programas de excelente calidad todos en U$ 100

Los cuatro programas, un procesador de textos base de datos planilla electrónica y un paquete gráfico son por separado comparables a cualquier otro software de su tipo En conjunto sin embargo tienen la capacidad adicional de la portabilidad de datos entre las distintas aplicaciones

Matsushita presenta el primer computador TRON

A tres años del inicio del programa TRON en Japon para desarrollar una nueva arquitectura de computadores Matsushita presentó al publico su primer prototipo mostrando que Japon está verdaderamente interesado en convertirse en una seria alternativa en computadores de 32 bits Para esto diversas empresas japonesas han unido sus esfuerzos en investigación y desarrollo

TRON son las siglas de 'real-time operating sistem nucleus incluye una arquitectura optimizada para una mayor velocidad y sistemas operativos distintos para aplicaciones distintas Es el caso de BTron (Business Tron) ITron (dedicado a robotica y sistemas de control) y CTron dirigido a mainframes

El objetivo japonés es crear un sistema alternativo a Unix y para ello han dotado a TRON aparte de su velocidad características de facilidad de manejo comparables a las desarrolladas en el sistema operativo del Macintosh

Una de las cualidades de TRON es que es independiente de la máquina en que se ejecute la interfaz con el usuario es comun en las diversas aplicaciones e incluye capacidad de comunicaciones

Del punto de vista hardware el equipo de Matsushita destaca por la ergonomia del teclado el que incluye tableta digitalizadora y scanner Su resolución es de 864 por 648 pixeles usa discos de 3 5 y viene con dos mega en RAM e interfaz para disco óptico Se espera que este prototipo ya salga a la venta a comienzos de 1989

Redes de Computadores

Con una masiva asistencia se desarrollo el seminario 'Redes de Comunicación organizado por Sonda y con la asistencia de ejecutivos de Digital Equipment Corp

El seminario, realizado en el hotel Carrera permitió a Sonda mostrar los productos Digital que la han puesto a la vanguardia en lo que a interconexión de computadores se refiere

La filosofía Digital de computación se ha caracterizado por su concepto de el sistema es la red donde los usuarios utilizando estaciones de trabajo se conectan a equipos centrales grandes los que configurados en redes permiten compartir en forma flexible los recursos computacionales

Es precisamente esa compatibilidad y conectividad entre los equipos de toda la linea Digital lo que se mostró en ese seminario destacando la compatibilidad de DNA cuyos productos pueden ser adquiridos aquí y ahora con los estándares que está proponiendo la ISO (International Standards Organization) para que sean adoptados por todos los fabricantes Esta compatibilidad es la que permite tambien a los equipos Digital conectarse a equipos de otros fabricantes

Seminarios en Logica

Un seminario dirigido a ejecutivos del área confección, calzado y textil realizó Logica recientemente Este seminario de cuatro sesiones sirvió para presentar el sistema MAI Basic Four para la industria de la moda

Este sistema, que es usado en más de 1 700 empresas en el mundo, fue presentado junto a testimonios de empresarios de Estados Unidos y México

En otro orden de cosas Logica organizó un seminario llamado Presente y Futuro de los Sistemas de Entrada de Datos en el cual fueron presentadas soluciones de Scan-Optics (Pertec)

Contadores Auditores

Durante los días 5 y 6 de junio se desarrolló en el Hotel Carrera la 1ª Convención Nacional de Contadores Auditores Universitarios de Chile

Este evento contó con el auspicio de ST Computación y permitió a la empresa organizadora presentar las herramientas computacionales más modernas que se utilizan en este sector

IMPRESORAS DE PRIMERA LINEA



Av. Pedro de Valdivia 1667 – Fono 2235721 – Casilla 130 Correo 29 – Providencia – TLX 340062 RIMPEX CK – Santiago

En el mercado el B38

Un nuevo componente de su familia de microcomputadores multiusuarios B25 comenzó a distribuir Unisys en Chile. El nuevo módulo procesador se denomina B38 y utiliza el microprocesador 80386 de reciente introducción en el mercado mundial.

Al mismo tiempo Unisys anunció mejoras a las capacidades de integración entre sus líneas de procesadores PC estándar y los microcomputadores B25, aumentando de esta manera el poder y la versatilidad de la familia B25.

El B38 puede administrar un cluster de hasta 11 estaciones de trabajo, sin restricción de combinaciones, siendo el mismo B38 una estación de trabajo de alto rendimiento. Opera bajo control del sistema operativo BTOS versión 8.0, lo que asegura su total compatibilidad con todas las familias de procesadores y periféricos de la familia B25.

Esta estación de trabajo existe en dos versiones: B38 CPU y B38 MCP. Esta última contiene todas las capacidades del primero más un coprocesador aritmético Intel 80287, operando a 10 MHz. La memoria en ambos modelos puede expandirse hasta 4 MB utilizando cartridges de un MB cada uno.

Instituto de Sistemas IBM

Durante tres días IBM reunió en Viña del Mar a casi un centenar de ejecutivos de informática de las principales empresas e instituciones del país, actuando como conferencistas expertos de IBM de distintas sedes latinoamericanas.

Este seminario estaba dirigido a mostrar un panorama completo de la tecnología actual de IBM. Para este efecto, se instaló un equipo de la serie 4300. Entre los temas tratados destacan Estrategias y direcciones de IBM, Tendencias en Sistemas grandes, Perspectivas en Comunicaciones, Bases de Datos Relacionales, Aplicaciones Avanzadas de Impresión.

Dentro de lo más resaltante de la estrategia de IBM debemos mencionar su interés en aumentar la productividad de la programación de aplicaciones a través de la nueva Arquitectura de Sistemas de Aplicación (SAA), la que permitiría crear aplicaciones portables de un miembro a otro de la familia de computadores IBM.



Sabemos todos los secretos de los nuevos Sistemas Personales/2 de IBM.
Este momento lo hemos preparado con tiempo Concretamente desde abril, cuando IBM nos informó de la llegada de sus nuevos Sistemas Personales/2. Desde entonces, estudiamos todos sus detalles y aplicaciones. Y hoy, que ya están en ST Computación, la empresa que más sabe de IBM en Chile, podemos revelarle todos los secretos y ayudarle a elegir el modelo que mejor responde a las necesidades de su empresa.
ST computación
Visítenos en nuestro edificio ST
Génova 2086 o llámenos al Fono 2514571
IBM
IBM

Taller de Ingeniería

Alrededor de 120 trabajos se expondrán en el X Taller de Ingeniería de Sistemas que se efectuará en forma conjunta con Softel 87 entre el 3 y 7 del próximo mes en el Hotel Crowne Plaza

El Taller, organizado por el Departamento de Ingeniería Industrial de la Universidad de Chile con la colaboración del Departamento de Sistemas de la Universidad Católica viene realizándose en torma anual desde 1977 Se trata de un evento tradicional dentro del medio profesional y académico chileno que paulatinamente ha ido incorporando participantes venidos de otros países latinoamericanos

Software municipal

Latindata desarrolló un software al que llamó Proyecto Multiusuario Area Municipal Este programa contiene paquetes, como licencia de conducir permiso de circulación, sistema de remuneraciones POJH y PEM fichas CAS, sistema de contabilidad gubernamental y otros

Los paquetes fueron desarrollados bajo sistema operativo Xenix y base de datos Relacional Informix

El proyecto considera tres fases fundamentales asesoría en terreno instalación del equipamiento adecuado y capacitación

Los primeros usuarios del Proyecto son las Municipalidades de Providencia y La Pintana

Curso CAD en U. de Concepción

Del 27 al 31 de julio será dictado en la U de Concepción un curso llamado "Introducción a los sistemas de diseño asistido por computador", el cual será dictado en español por el Dr Pierre Beckers del Laboratorio de Técnicas Aeronáuticas y Espaciales de la Universidad de Lieja, Bélgica

Paralelamente se exhibirán equipos para el uso profesional de los sistemas CAD, como plotters digitalizadores, computadores y otros El curso cuenta con la aprobación del SENCE, código 06-05-0310-08 Mayores informaciones se pueden solicitar a los organizadores, Télex 260157 INCON CL o al teléfono 234985, anexo 2327. Concepción

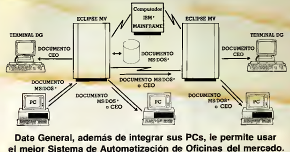
Microbyte en Argentina

Gran éxito tuvo la presentación de nuestra revista en la mayor exposición que se realiza en Argentina en el área computación y telecomunicaciones

Infotelecom 87, se llevó a cabo los primeros días de junio y ahí estuvieron presentes alrededor de 100 de los principales proveedores de hardware, software y servicios del mercado argentino En conjunto, se realizó además el congreso de Usuaria en la cual se dictaron cerca de 300 conferencias con la asistencia de unos 10 000 participantes

Se calcula que asistieron a la muestra unas 50 000 personas, Microbyte estuvo presente en un stand en el cual pudo estrecharse aun más los lazos que nos unen con los lectores argentinos luego de haber editado para ese país ya 8 ediciones, a contar de septiembre pasado Cabe destacar que en Argentina, Microbyte ha logrado un gran éxito y es la única revista tecnico profesional que circula Todas las otras revistas apuntan al segmento computadores hogereños

¿UD. Y SU PC, SE SIENTEN AISLADOS DENTRO DE SU EMPRESA?



Data General, además de integrar sus PCs, le permite usar el mejor Sistema de Automatización de Oficinas del mercado.

Data General ha desarrollado un sistema que permite integrar la funcionalidad del editor de textos o de la planilla electrónica de su PC al sistema de automatización de oficinas CEO, considerado como el paquete más avanzado de automatización de oficinas.

Este producto, llamado CEO CONNECTION, permite que su PC IBM* compatible se transforme, al seleccionar alternativas de un menú, en una estación de trabajo del sistema CEO o del sistema ECLIPSE MV, o seleccione editores de texto o de planillas electrónicas propias del ambiente MS/DOS* de su PC.

Bajo CEO CONNECTION, residente en su PC y en el sistema ECLIPSE MV, Ud puede seleccionar ambas modalidades, consiguiendo integrar estos dos ambientes. Data General le ofrece adicionalmente un avanzado software de procesamiento de texto para su PC, CEOwrite, el que permite ingresar directamente al ambiente CEO del computador central.

Adicionalmente, y producto de esta integración, CEO CONNECTION y CEO le permitirá

- Editar, almacenar en el sistema CEO del sistema central y enviar por correo electrónico, documentos de texto de Wordstar*, Wordperfect*, Multimate*, IBM DCA* (en la forma revisable o final) y CEOwrite, a otros usuarios de PCs
- Editar, almacenar en CEO y enviar por correo electrónico documentos de toma de decisión, tales como Lotus 1-2-3* y otros que soporten el formato DIF, a otros usuarios de PCs
- Enviar, almacenar y recuperar archivos binarios MS/DOS* bajo el sistema operativo del sistema ECLIPSE MV.
- Convertir documentos de texto o de planillas electrónicas, creados con los sistemas mencionados anteriormente, al formato CEO y viceversa.



* Marca Registrada



Inauguran red donada por IBM

Una red local compuesta de 20 microcomputadores donó IBM al Departamento de Ingeniería Industrial de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile La entrega forma parte de un nuevo 'Programa Académico Conjunto entre ambas instituciones El costo comercial de la donación alcanza a 100 mil dólares

La inauguración de los equipos y el proyecto tuvo lugar en junio recién pasado ocasión en que hicieron uso de la palabra Hernán Cervallo gerente general de la empresa de computación y Antonio Holgado a nombre del Departamento

Cervallo destacó en su intervención el rol que el Departamento juega en la formación y capacitación de ejecutivos profesionales y empresarios A continuación anadió que 'esto permite asegurar que el desarrollo de este programa producirá en el futuro una efectiva transferencia de tecnología a la comunidad empresarial'

Los equipos (2 AT y 18 XT) de 512 k están interconectados mediante la red PC Network El software de la red permite que se puedan compartir los recursos de todos y cada uno de los equipos interconectados tales como el almacenamiento en disco duro y el uso de las impresoras La red se encuentra instalada de tal manera que algunos microcomputadores móviles pueden instalarse en las salas de clase en las oficinas de los academicos y en los puestos de trabajo de los administrativos Desde cualquiera de ellos se puede cargar un programa de uso corriente enviar mensajes de correo electronico a otro equipo comunicarse con los computadores de las instalaciones del centro de computación y a futuro con las lineas de conexión a redes universitarias extranjeras



Hernán Cervallo gerente general de IBM recordó los 30 años de colaboración de su empresa con la Universidad de Chile

Coasin ofrece nuevo modem

Un modem con una nueva tecnología comenzó a comercializar Coasin Es el RM 9632 de Racal-Milgo que permite transmitir datos full duplex sobre la red telefónica publica a 9 600 y 4 800 bps Opera bajo la nueva forma V32 con codificación Trellis del CCITT

Dentro de sus características de acción incluye autodiscado operación no atendida con respuesta automática y ajuste automático de velocidad

La modulación con codificación Trellis permite un mejor rendimiento a altas velocidades de transmisión sobre las líneas de las redes de teléfonos

Además de operar sobre las líneas conmutadas, lo hace sobre las líneas dedicadas de dos o cuatro hilos

Este modem da una buena solución en aplicaciones donde se requiere hacer transmisiones masivas de datos a lugares en los que no se dispone de líneas dedicadas o se requiere hacer transmisión de datos a altas velocidades en formas esporádica

Softel 87 a la vista

Softel 87 anuncia la utilización de un concepto diferente de exposición. En él convergen originales ideas sobre medios de venta, comunicación y publicidad. El objetivo es proporcionar un ambiente adecuado a las empresas de informática y telecomunicaciones para presentar a un segmento seleccionado del mercado, todos los avances tecnológicos que ellas ofrecen.

La conocida convención de informática es organizada por la Fundación de Estudios Prospectivos de la Universidad de Chile. Se espera la participación de alrededor de 50 proveedores de hardware y software. El lugar del evento es el mismo de los anteriores: el Hotel Crowne Plaza de Santiago. La fecha: del 3 al 7 de agosto próximo.

Además de la muestra, Softel 87 presenta el XI Encuentro Latinoamericano de Usuarios de Informática y Telecomunicaciones y el Primer Ciclo de Paneles y muestras prácticas de resultados de empresas usuarias nacionales en informática y telecomunicaciones.

El Encuentro comprende conferencias y paneles. Se inaugura el 4 de agosto en el Edificio Diego Portales, donde se desarrollará íntegramente. Dictarán las conferencias Ruth Donoso y Eugenio Bonnefont. La primera disertará sobre "Enseñanza media". Bonnefont por su parte hablará sobre "Los servicios de transmisión de datos y su comercialización".

También en el Diego Portales tendrá lugar el I Ciclo de Paneles entre el 3 y el 7 de agosto. Para su realización se han definido cinco áreas: financiera, previsión y seguros, productiva, minería y agrícola forestal.

La programación general de cada área comprende presentación de una empresa con aplicaciones computacionales del área específica, presentación de una empresa proveedora de aplicaciones en el área, mesa redonda y, por último, presentación de otros proveedores sobre aplicaciones específicas.

El XI Encuentro lo organiza la Federación Latinoamericana de Usuarios de Informática y Telecomunicaciones (FLAI) y el I Ciclo de Paneles la Corporación de Empresas e Instituciones Usuarias de la Informática y las Telecomunicaciones (ASEUCI). Esta última, de reciente creación, representa en Chile a la primera.

Mayores informaciones sobre estos eventos pueden solicitarse a los teléfonos 2315594 y 2323609.

Mainframes de escritorio

Por unos dos mil dólares más el costo de un IBM PC o un compatible, es posible tener en el escritorio el poder computacional de un verdadero mainframe

Este es el resultado de la incorporación de tarjetas MicroWay que contienen dos megabytes de RAM y el Transputer, el sistema de procesadores en paralelo desarrollado por Inmos en Gran Bretana Usando el Transputer, el ciclo del computador se acelera hasta 20 MHz casi cinco veces la velocidad normal de un PC lo cual le permite trabajar a alrededor de 10 mips (millones de instrucciones por segundo), lo mismo que el 3081, el mainframe introducido por IBM hace sólo seis años El procesador 8088 original del PC queda relegado a tareas de administración de recursos pantalla y teclado

El unico problema en esta verdadera maravilla es que no existe ningun software que sepa aprovechar todo ese potencial Sin embargo a pesar de lo anterior estas nuevas tarjetas han tenido una gran venta, sobre todo de parte de usuarios que requieren de esa impresionante velocidad para cálculos matemáticos y por gente que necesita experimentar en procesamiento paralelo

Junto con la tarjeta el comprador recibe un sistema de desarrollo para Occam el lenguaje de programación desarrollado por Inmos para el Transputer Otros lenguajes de proxima aparición desarrollados por Lattice Logic de Edimburgo son compiladores para el Transputer de C Pascal y Fortran 77

Congreso Iberoamericano de Inteligencia Artificial

En Barcelona, España entre el 11 y 13 de enero de 1988 se llevará a cabo el Primer Congreso Iberoamericano de Inteligencia Artificial (IBERAMIA 88) patrocinado por las asociaciones española portuguesa y mexicana de inteligencia artificial

Para este efecto los organizadores han llamado al publico iberoamericano a participar enviando articulos originales relacionados con la Inteligencia Artificial con un maximo de 20 páginas y un resumen de 500 palabras a Secretaria Iberamia Feria de Barcelona Av Reina Mª Cristina s/n 08004 Barcelona España

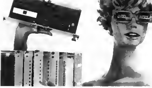
Paralelamente al congreso tendrá lugar una exposición de material informático y de software

Mercado mundial de la Inteligencia Artificial

De acuerdo a estudios realizados por el Bureau d Information et de Previsions Economiques (BIPE) el mercado de la inteligencia artificial concentrado abrumadoramente en los Estados Unidos estaria alcanzando un nuevo equilibrio con la paulatina integración de Japón y Europa

Un factor interesante en este estudio es la relación existente entre hardware y software en el mercado de la I A la que estaria revirtiéndose en favor del software especialmente en desarrollo de sistemas expertos aprendizaje y representación de conocimientos

En 1990 se estima que el mercado estará dividido en un 64% para Estados Unidos (72% hoy) contra un 19% en Europa y 17% Japón El valor de éste ascenderia de U$ 700 milones en 1986 a U$ 4 000 millones en 1990



Auditoria Informática

Con el auspicio del CREI (Centro Regional del IBI para la Ensenanza de la Informática) se llevará a efecto en Puerto Rico entre el 2 y 6 de noviembre del presente, un seminario sobre Informatica y Auditoria

Además de las ponencias preparadas y encargadas expresamente por los organizadores a personas de reconocido prestigio se invita a presentar comunicaciones escritas sobre los temas auditoria factor riesgo seguridad de instalaciones metodologia tecnicas y tendencias

Los trabajos deberán tener un maximo de 15 páginas y el resumen no más de 150 palabras El plazo de recepción es hasta el 31 de julio en la Secretaria del Congreso P O Box 21869 UPR Station San Juan Puerto Rico 00931

Mitac libera super AT

Mitac Int. Corp., uno de los mayores fabricantes de computadores en Taiwán, anunció la pronta liberación de su propio equipo super AT, los poderosos microcomputadores que operan basados en un procesador 80386.

Junto con Mitac, ya son muchos los fabricantes que han lanzado sus propios super AT, en lo que va corrido de este año, sin embargo el software operativo para sacar beneficio de estos monstruos tarda en aparecer.



Computadores desocupan las cárceles

Es muy común en prácticamente todos los países, que los sistemas judiciales se vean enfrentados a la contradicción de infringir la propia legalidad al tener las cárceles colmadas en condiciones incluso infrahumanas por falta de mayor espacio. Al mismo tiempo el presupuesto invertido por el estado para la mantención y eventual rehabilitación de los reclusos nunca es suficiente y menos aun recuperado de ninguna forma.

El ingenio de fabricantes norteamericanos estaría sin embargo aportando una solución a este mal. En efecto en algunos estados se ha instituido una forma de arresto domiciliario electrónico que es ventajoso también para los reclusos.

El sistema de vigilancia electrónica es producido por BI Incorporated, de Boulder Colorado y consiste en una pulsera tipo esposas que transmite permanentemente una señal la que es captada aleatoriamente por un receptor en un aparato telefónico, el cual transmite la información a un computador central, el que lleva así el control de donde está cada uno de los 200 prisioneros que es capaz de supervisar. El computador puede ser programado para admitir la no recepción de mensajes durante determinadas horas para así permitir al recluso trabajar y llevar una vida casi normal.

La tasa de reincidencia en este tipo de reclusos llegó a sólo el 19% comparado con el 33% de reincidentes en sistemas de libertad condicional tradicionales. Al mismo tiempo, el estado ahorra el costo de US$ 32 dólares diarios que gasta diariamente por prisionero. Como dato de interés cabe destacar que BI Incorporated es la misma que diseñó unos equipos similares que son colgados en el cuello de las vacas en un establo y que permiten monitorear lo que comen y también lo que producen para así ir ajustando las raciones alimenticias.

Procesamiento paralelo

Sin duda, los grandes computadores de cinco años más serán muy distintos a los que conocemos hoy, y la razón son los avances en procesamiento paralelo.

De acuerdo a un estudio publicado en Computer Economics, los computadores en 1992 correrán a mil gigaflops (billones de instrucciones en punto flotante por segundo) y a un costo significativamente inferior al actual.

Para esto se han desarrollado en lo fundamental dos tendencias en la construcción de procesadores paralelos. La primera estrategia conocida como farm, utiliza unos pocos procesadores en general menos de ocho, los que comparten y se comunican a través de una memoria común. En general el software de estos equipos es fácilmente transportable desde ambientes standard.

La segunda estrategia conocida como "cubos", utiliza un número mayor de procesadores, aunque de una menor capacidad. En este caso la relación costo-beneficio es mejor dada la accesibilidad de procesadores y al hecho de que para aumentar la capacidad de un equipo basta con incorporar nuevos procesadores en lugar de rediseñar y desarrollar procesadores más complejos y poderosos.

Si bien estos desarrollos se prevén principalmente en el área de supercomputadores, pues los mainframes son más especializados en tareas de input-output, el otro segmento en el que el paralelismo hará su entrada es el de minicomputadores de corte ingenieril y de diseño y fabricación asistidos por computador (CAD-CAM).

Todo programa puede ser estructurado siguiendo el método que presentamos a continuación.

PROGRAMACION ESTRUCTURADA

Mercela Rome

Algunos lenguajes de programación se prestan perfectamente a las necesidades de la programación estructurada mientras que otros no se adaptan tanto Esto está justificado pues los lenguajes que aparecieron y se desarrollaron antes de la llegada de la programación estructurada permiten estructuras de control no conformes a las reglas de ésta mientras que los aparecidos y desarrollados paralelamente a la programación estructurada como el Pascal se adaptan perfectamente a su realización

Un programa estructurado de cualquier tamaño puede ser leído y entendido de un modo sistemático leyendo e interpretando la estructura jerárquica de los programas primos que lo compone y sus abstracciones

Los objetivos principeles de la lectura de un programa estructurado son

–Verificar que el programa es correcto respecto de una función dada

- Descubrir la función de un programa dado

Para poder expresar un diseño de software de un modo formal y preciso y al mismo tiempo más general que la gramática de un lenguaje se utilizan los lenguajes de diseño de programas (PDL Process Design Language) en los que aparecen dos estructures sintácticas

–Sintaxis interna

Se define todo aquello que no depende del lenguaje particular estructuras de control de datos y de sistemas

–Sintaxis externa

Se definen los tipos de datos y las operaciones con ellos

Cualquier estructura compleja puede representarse con las estructuras de control del PDL (secuencia, decisión e intereccción) las cuales no tienen transferencia incondicional (GOTO) A su vez si se restringen las estructuras de control a utilizar, pueden formarse distintes bases para expresar programas

"TEOREMA DE ESTRUCTURACION DE MILLS"

El teorema de la estructuración o de Mills permitirá convertir cualquier estructura compleja en una estructura equivalente que utilice sólo estructuras del PDL En este teorema se afirma que cualquier programa propio es funcionalmente equivalente a un programa estructurado que utiliza como base el siguiente conjunto secuencia IF THEN ELSE WHILE DO manteniendo los nodos función y predicados originales y utilizando un contador e instrucciones de asignación adicional

Pasos para estructurar un programa no estructurado

1 - Enumerar todos los nodos función y predicado del programa de 1 a N asignando al nodo de salida el numero 0 (cero)

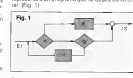
2 - A la salida de cualquier nodo función o predicado asignar a un contador auxiliar L el numero correspondiente al nodo de destino

3 - Construir un programa formado por

- El valor inicial de L en 1
- Un lazo WHILE DO donde dentro del DO hay una sucesión IF THEN ELSE testeando el valor de L (La sucesión puede reemplazarse por un CASE)
- Si L = i en el THEN correspondiente se ejecuta el bloque Gi del programa original
- Si WHILE DO termina por L = 0

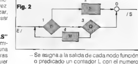
Apliquemos el teorema en un ejemplo

Supongamos tener un diagrama de flujo que represente a un programa que se desea estructurar (Fig 1)

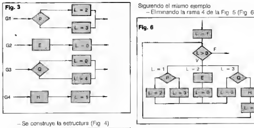


Pasos

– Se numeran los nodos (Fig 2)

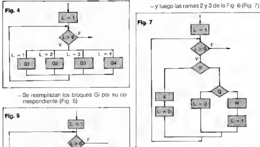


– Se asigna a la salida de cada nodo función o predicado un contador L con el numero correspondiente al nodo de destino (Fig 3)



– Se construye la estructura (Fig 4)





– Se reemplazan los bloques Gi por su correspondiente (Fig 5)

Fig. 5

Esta forma puede tener redundancias por lo que se admite una simplificación que lleva a lo que se llaman estructuras recursivas y que consiste en reemplazar la aparición de la asignación al contador en una rama j por la rama distinta de j correspondiente

Siguiendo el mismo ejemplo

– Eliminando la rama 4 de la Fig 5 (Fig 6)

– y luego las ramas 2 y 3 de la Fig 6 (Fig 7)

Con lo que se obtiene un programa estructurado a partir de uno no estructurado (Compárese la Fig 1 con la Fig 7)

En general, cuando se tiene un programa no estructurado conviene extraer los programas primos y luego tratar solamente el subprograma no estructurado Este procedimiento se conoce como "PARSING de programas primos", donde puede ocurrir que aparezcan estructuras que sean primas pero que no pertenezcan al subconjunto que se ha tomado como base en este caso se sigue con el parsing hasta llegar a un solo bloque función en el que el número de nodos debe coincidir con la suma de nodos predicado y función del diagrama original En este punto hay que concen-

Continúa en pág 33


COMPUTER CLUB
Computer Club es una sección escrita fundamentalmanta por ustades los lectores En éste se incluyen todos los aspectos de la microcomputación, dasda programas da juegos, utilitarios a programas administrativos para todos los microcomputadores
Los programas a publicar pueden ser an Basic, código de máquina o cualquier otro, pero al enviar su colaboración asegurese de
■ acompañar un cassette o disco para verificar al buen funcionamiento de su programe
■ incluir una breve dascripción de qué as lo qua hace al programa y cómo
• en lo posible incluir un listado por impresora El listado debe sar claro como para reproducirlo, si su cinta no es nueva, imprima enfatizado
• qua los caracteras gráficos o an video inverso aparazcan claramente en el listado o de lo contrario incluya líneas REM dascribiéndolos
Todas las colaboraciones publicadas serán pagadas a $ 3 000 o ▲30
Envie sus colaboraciones a:
Computer Club
Revista Microbyte
Huelen 164 - 2° piso
Santiago
o
Viamonte 723,
2° P. Of. 7
Buenos Aires
Cap Fed
22 Rincón Commodore
25 Spectrum :
26 Casio :
28 Atari :
32 Commodore :

El rincón de Commodore

Eduardo Ahumada M.

Este mes trataremos el tema de la gestión de datos usando el archivo interno DATA del BASIC Commodore, en ediciones posteriores veremos el uso de Archivos en cinta magnética y en diskette.

Básicamente, el archivo DATA es un archivo secuencial que sólo puede usarse en modo de lectura, es decir el programa puede recuperar información de este archivo, más no puede modificarla. Para cambiar un ítem de datos del archivo DATA, es necesario que el operador modifique el programa, alterando las sentencias DATA correspondientes, y luego guarde el programa así alterado en lugar del programa original.

Existe una técnica que permite que un programa se modifique a sí mismo, con lo cual se podría hacer que el programa agregara información al archivo DATA, pero este método es bastante complejo, de modo que es preferible ceñirse a lo que permite el lenguaje BASIC en forma estándar.

A cambio de esta limitación, el archivo DATA nos ofrece una gran velocidad de acceso secuencial a la información, mucho mayor que la que puede lograrse en un archivo secuencial en cassette o diskette.

Estas dos características, capacidad sólo de lectura y rapidez de acceso secuencial, hacen que este tipo de archivo sea ideal para la implementación de bases de datos del tipo de consulta por contenido. Como ejemplo veremos el típico caso de una Base de Datos Discográfica, en la que tendremos registrado el contenido de nuestra discoteca. Implementaremos paso a paso un programa de consulta, el cual deberá cumplir las siguientes funciones:

a) Búsqueda por intérprete.
b) Búsqueda por nombre de canción.
c) Las búsquedas serán del tipo "calce parcial", por ejemplo, si se desea buscar las canciones de Stevie Nicks, pero no estamos seguros de cómo se escribe su apellido, podemos buscar por Stevie y el programa nos encontrará las canciones de Stevie Wonder, Stevie Nicks, John Stevie, etc.

Para diseñar este tipo de programas, lo primero que debe hacerse es definir el formato que tendrá la base de datos. Cada registro contendrá la información de un "volumen" (Disco o Cassette), según el formato siguiente:

```
DATA  Número de Volumen, Título del Disco
DATA  *, Intérprete 1, Nombre de la canción 1
DATA  *, Intérprete 2, Nombre de la canción 2
 ..   *    ..          ..
 ..   *    .           ..
```



Este formato se repetirá tantas veces como volúmenes tenga nuestra discoteca. Veamos un ejemplo de registro de esta base de datos:

```
DATA  001, Fleetwood Mac, Live
DATA  *, Fleetwood Mac, Sara
DATA  *, Fleetwood Mac, Over & Over
DATA  *, Fleetwood Mac, Never Make me Cry
 ..   *
(etc.)
```

Para ahorrar memoria, emplearemos un pequeño truco: Si hay varias canciones consecutivas del mismo intérprete (como usualmente será el caso), podremos abreviarlo con el símbolo "=", que significará "el mismo intérprete de la canción anterior", por ejemplo:

```
DATA  001, Fleetwood Mac, Live
DATA  *, Fleetwood Mac, Sara
DATA  *, =, Over & Over
DATA  *, =, Never Make me Cry
 .    *, =,
(etc.)
```

La tentación de ahorrar aun más memoria, colocando el nombre del intérprete junto a los datos del volumen (primera línea) debe ser resistida, pues este esquema es más flexible, permitiendo el registro de cassettes que contengan canciones de varios intérpretes.

Además de definir el formato de los registros de nuestra Base de Datos, debemos idear un método para que el programa puede detectar el Fin de Archivo, es decir el punto en el cual terminan nuestros datos. La forma más fácil de lograrlo es colocando un registro "falso", que tenga un código de Volumen especial que el programa reconozca como el fin de los datos. Arbitrariamente fijará que el Volumen con el código "-FDA-" será el Fin de Archivo.

Analicemos un instante el uso de espacio que requerirá este archivo. Podemos suponer que en promedio cada canción usará unos 26 bytes, que cada volumen tendrá unos 10 canciones, y que para registrar los datos del disco usaremos unos 30 bytes adicionales, lo cual arroja unos 290 bytes por cada disco que registra la base de datos. En el C-64 hay unos 39 000 bytes disponibles, reservemos 5 000 para el programa y sus variables, ello nos deja 34 000 bytes para las sentencias data. Esto es suficiente para unos 117 volúmenes, lo cual es razonablemente grande para una colección casera.

Como todos sabemos, un programa BASIC de 39000 bytes es un GRAN programa, y la Unidad de Diskette tardaría unos 3 minutos en cargar o memoria este pedazo de programa (39000 bytes = 154 bloques de diskette), y más de 20 minutos en cargar desde cassette. Este problema tiene dos soluciones posibles: usar un utilitario que acelera la carga de programas, tales como el TURBODISK o TURBOTAPE de la Compute Publishing Co., o dividir nuestra Base de Datos en subconjuntos más pequeños y disjuntos.

Por ejemplo, supongamos que tenemos una colección de 200 LPs, que se pueden clasificar en la forma siguiente:

80 LPs de Rock Moderno
30 LPs de Jazz
60 LPs de Música Clásica
30 LPs de Musica Folclórica

En un diskette se podría tener cuatro versiones del programa, en la forma siguiente:

INDICE ROCK	111 Bloques
INDICE JAZZ	54 Bloques
INDICE CLASICA	88 Bloques
INDICE FOLK	54 Bloques
Total	307 Bloques (46% de un Diskette)

Hagamos un resumen de este diseño: Ya hemos definido lo que deseamos obtener del programa, diseñamos la configuración de los registros de la Base de Datos y estudiamos las limitaciones y capacidades máximas de almacenamiento de datos. A continuación comenzaremos la implementación del programa propiamente tal.

En programación, al igual que en el karate, hay diversas escuelas o "estilos". Mi preferencia particular es el estilo "Bottom-Up", que preconiza el atacar primero las funciones más complejas.

Indudablemente, la función más básica del programa es que al buscar por Intérprete o por tema, lo hago con el método de "calce parcial", es decir, necesitamos una función para buscar una cadena de caracteres dentro de otra, de tal manera que si tenemos las variables siguientes:

```
IN$ = "Stevie Nicks"
X$ = "Nicks"
```

el programa se dá cuenta de que el argumento de busqueda X$ "calza" con el Intérprete IN$. El BASIC 7.0 del Commodore 128 posee una función intrínseca que proporcione esta facilidad: la función INSTR. Es interesante que estudiemos su sintaxis, pues nos ayudará a definir una función similar para el BASIC 2.0 del C-64. El formato de esta función es:

```
INSTR (X$, Y$, Z)
```

Donde Y$ es la cadena que se deberá buscar dentro de la cadena X$, a partir de la posición Z de X$.

La función entrega la posición de comienzo de Y$ dentro de la cadena X$, o un cero en caso de que Y$ no está contenido en X$. Veamos algunos ejemplos:

```
INSTR ("Stevie Nicks", "Nicks", 1) entrega un 8
INSTR ("Stevie", "Nicks", 1)     entrega un 0
```

En nuestro caso, sólo nos interesa saber si Y$ está o no contenido en X$, por lo cual fijaremos el valor Z en uno. Una posible implementación de esta función en BASIC 2.0 es:

```
37000 REM * Función INSTR *
37005 :
37010 FOR P=1 TO (LEN (X$)-LEN (Y$) +1)
37015 : IF MID$ (X$, P, LEN (Y$) )=Y$ THEN
      RETURN
37020 NEXT: P=0: RETURN
```

Esta implementación funciona correctamente en la mayoría de los casos, pero puede fallar si el largo de Y$ es mayor que el largo de X$. Una versión mejorada (y que además ejecuta más rápido, pues las operaciones repetidas se han eliminado), pero que requiera las variables auxiliares I9 y J9, es la siguiente:

```
800 :
810 REM * Función INST *
820 : I9=LEN (Y$): J9=LEN (X$)-I9+1
```

```
825  IF J9 <1 THEN P=0 RETURN
830  FOR P=P TO J9
835  IF MID$ (X$, P, I9) = Y$ THEN RETURN
840  NEXT P=0 RETURN
```

Aunque esta versión es algo más larga, ejecuta varias veces más rápido, pues la comparación de la línea 835 es más simple de llevar a cabo por parte del intérprete BASIC

Supongo que al lector atento le parecerá extraño que haya variado los numeros de línea, del 37000 al 800 La razón está en que incluso antes de escribir un programa, es necesario tener una idea de la distribución que se dará a los números de línea Para este programa he decidido dejar las líneas 1-299 para el cuerpo principal del programa (un Menu que permitirá llevar a cabo tres funciones. Listar la Base de Datos completa, Buscar por Intérprete, y Buscar por Canción), las líneas 300-499 para la Opción 1 del Menú, 500-699 para la Opción 2, 700-799 para la Opción 3, 800-1999 para las funciones básicas, y de la 2000 en adelante para almacenar las sentencias DATA de la Base de Datos

El estilo de programación "Bottom Up" requiere que las funciones se prueben a medida que se implementan Para probar la rutina anterior usé el siguiente trozo de código:

```
10 INPUT "X$, Y$"; X$, Y$
20  GOSUB 800 PRINT "P="; P
30 GOTO 10
```

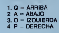
Bueno, el espacio se me está acabando, por lo que continuaremos con el programa el próximo mes, pero para que se entretengan mientras tanto, pueden comenzar a ingresar al computador los datos de sus Discotecas, siguiendo el ejemplo que aparece en el listado 1

```
2000 '
2010 rem' * Base de Datos *
2020
2030 data 001,fleetwood mac live
2040 data *,fleetwood mac,sara
2050 data *,=,over & over
2080 data *,=,rhiannon
2070
2080 data 002,queen  greatest hits
2090 data *,queen,bohemian rhapsody
2100 data *,=,bycicle race
2110 data *,=,fat bottomed girls
2120
2130 data 003,musica en espanol
2140 data *,jose luis perales,por amor
2150 data *,=,y como es el?
2160 data *,miguel bose,marchate
2170 data *,julio iglesias,por el amor de una mujer
2180 '
2190 data 004,rock argentino i
2200 data *,soda stereo,nada personal
2210 data *,=,cuando pase el temblor
2220 data *,=,persiana americana
2230 data *,=,telequinesis
2240 data *,virus,imagenes paganas
2250 data *,=,sin disfraz
2260 data *,leon gieco,solo le pido a dios
2270
2280 data 005,rock chileno
2290 data *,aparato raro,restricciones televisivas
2300 data *,los prisioneros,mueven las industrias
2310 data *,=,baile de los que sobran
2320 data *,engrupo,historia
2330 data *,=,me tranquilizare
2340 data *,los socios,desde las nubes
2350 data *,numero equivocado,bonnie & clyde
60000 :
60010 data <fda>
60015 .
60020 print clear$

ready
```

Punk-Man

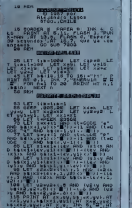
PUNK-MAN es una versión del conocido PAC-MAN para los usuarios de TIMEX 2048 que nos envía ALEJANDRO LAGOS SANTELICES. La utilización de este programa es realmente sencilla. Los controles son los siguientes



1. Q = ARRIBA
2. A = ABAJO
3. O = IZQUIERDA
4. P = DERECHA

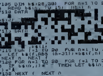
Además tras un selector de "grado de dificultad", el cual presenta tres niveles. El primer nivel, es el más sencillo de todos, ya que los perseguidores tienen sólo un tercio de la velocidad de la nave perseguida. Luego en el segundo nivel la velocidad de la nave enemiga aumenta a la mitad de la nave perseguida. Es así como se llega al tercer nivel, donde la nave enemiga tiene casi la misma velocidad de la nave perseguida.

Es por todo lo antes mencionado que estamos seguros que los usuarios de este juego lo harán uno de sus favoritos, puesto que goza de una gran dinámica y al mismo tiempo es fácil de manejar.



```
  10 REM
         ◆◆PUNK-MAN◆◆
          © 1987 por
        Alejandro Lagos
         STGO.,CHILE
  15 BORDER 0  PAPER 0  INK 4  C
LS   PRINT AT 5,11, FLASH 1,"PUN
K-MAN",AT 13,8, FLASH 0,"Espera
30 segundos",AT 15,7,"que ya com
enzamos"  GO SUB 7000
  20 REM
         ■VARIABLES■
  25 LET tim=1000  LET cap=0  LE
T limit=300  LET x=8  LET y=15
LET x1=19  LET y1=29  LET x2=2
LET y2=2  LET vid=3
  45 LET b$(10,15 TO 16)="M"  P
RINT AT 0,1, INK 7,"ENERGIA ■■
■"  FOR n=1 TO 20  PRINT AT n,1
,b$(n)  NEXT n
  50 REM
       ■■PARTE PRINCIPAL■■
  53 LET tim=tim-1
  55 BEEP ,005,20  LET xx=x  LET
 yy=y  LET xx2=x2  LET yy2=y2  L
ET yy1=y1  LET xx1=x1
  60 LET l=PEEK 23560
  73 LET x=x+(1 AND (l=CODE "a"
AND b$(x+1,y)<>"■"))-(1 AND (l=C
ODE "q" AND b$(x-1,y)<>"■"))
  77 LET y=y+(1 AND (l=CODE "p"
AND b$(x,y+1)<>"■"))-(1 AND (l=C
ODE "o" AND b$(x,y-1)<>"■"))
  85 LET x1=x1+((lev AND (x1<x AN
D b$(x1+lev,y1)<>"■"))-(lev AND
(x1>x AND b$(x1-lev,y1)<>"■"))
  90 LET y1=y1+((lev AND (y1<y AN
D b$(x1,y1+lev)<>"■"))-(lev AND
(y1>y AND b$(x1,y1-lev)<>"■"))
  95 LET x2=x2+((lev+.1 AND (x2<x
 AND b$(x2+lev+.1,y2)<>"■"))-((le
v+.1 AND (x2>x AND b$(x2-lev+.1,
y2)<>"■"))
 100 LET y2=y2+(.8 AND (y2<y AND
 b$(x2,y2+.8)<>"■"))-(.8 AND (y2
>y AND b$(x2,y2-.8)<>"■"))
 115 PRINT AT xx,yy,b$(xx,yy),AT
 xx1,yy1,b$(xx1,yy1),AT xx2,yy2;
b$(xx2,yy2),AT x,y, INK 6,"■",AT
 x1,y1, INK 6,"■",AT x2,y2, INK
6,"■"
 117 IF (x=INT (x1+.5) AND y=INT
 (y1+.5)) OR (x=INT (x2+.5) AND
y=INT (y2+.5)) THEN GO TO 2000
 120 IF b$(x,y)="*" THEN LET b$(
x,y)=" "  BEEP ,007,40  LET cap=
cap+1  IF cap=limit THEN GO TO 1
000
 150 GO TO 60
1000 REM
        ■■FINAL FELIZ■■
1005 FOR n=1 TO 25 STEP 4  FOR i
=0 TO 7  BEEP .02,(i+n)*2  BORDE
R i  NEXT i  NEXT n  BORDER 0
1010 BEEP .9,0  BEEP .9,20. FLAS
H 1  CLS
1013 PRINT AT 13,1,"ENERGIA=",("
■■■"(1 TO vid))
1015 PRINT AT 1,1, INK 4,"FELICI
TACIONES !!",AT 11,1,"PUNTAJE= "
  FOR n=0 TO tim  BEEP ,005,2  P
RINT AT 11,10,n  NEXT n
1017 PAUSE 100  FOR n=1 TO vid
PRINT AT 11,10,"000"  BEEP .01,0
  PRINT AT 11,10,tim+(100*n)  BE
EP .01,2  BEEP .01,0  PRINT AT 1
3,8+n," "  NEXT n
1020 INPUT "De nuevo (s/n)? ", L
INE z$  IF z$="n" THEN FLASH 0
CLS  STOP
1025 IF z$="s" THEN FLASH 0  RUN

1030 GO TO 1020
2000 REM
       ■■PERDIDA VIDA■■
2005 BEEP .5,0  BEEP .5,8  BEEP
1,-10
2010 LET vid=vid-1  PRINT AT 0,9
+(vid*2),"  "  IF vid<=0 THEN PA
USE 100  BEEP .2,0  BEEP .4,10
BEEP .5,0  BEEP .08,-10  BEEP 1,
-30  CLS  PRINT AT 11,0,"PERDIO
!!"  GO TO 1020
2015 PRINT AT x1,y1,b$(x1,y1),AT
 x2,y2,b$(x2,y2)  LET x2=2  LET
y2=2  LET x1=19  LET y1=29
2020 GO TO 60
7000 REM
        ■■GRAFICOS■■
7005 FOR n=USR "a" TO USR "c"+7
 READ a  POKE n,a  NEXT n
7010 DATA 129,0,66,90,24,60,126,
255
7015 DATA 36,36,60,255,153,0,255
 60
7020 DATA 0,0,0,0,50,126,0,0
7100 REM
        ■■LABERINTO■■
7105 DIM b$(20,30)  FOR n=1 TO 2
0  READ b$(n, TO 15)  NEXT n
7110 DATA
```



```
7115 FOR f=1 TO 20  FOR n=1 TO 1
5  LET b$(f,ABS (n-31))=b$(f,n)
 NEXT n  NEXT f
7120 FOR n=1 TO 20  FOR f=1 TO 3
0  IF b$(n,f)=" " THEN LET b$(n,
f)="*"
7130 NEXT f  NEXT n
7200 REM
      ■NIVEL DE DIFICULTAD■
7205 BEEP .9,40  CLS
7210 PRINT AT 0,0,"Entre nivel d
e dificultad(1/3) ",AT 5,0, FLAS
H 1,1,AT 11,0,2,AT 16,0,3, FLASH
 0,AT 5,2,"FACIL",AT 11,2,"NORMA
L",AT 16,2,"DIFICIL"
7215 INPUT lev  IF lev>3 OR lev<
1 THEN GO TO 7215
7220 LET lev=(lev+3)*.1  CLS
7500 RETURN
9999 SAVE "PUNK-MAN" LINE 0
```

Laberinto

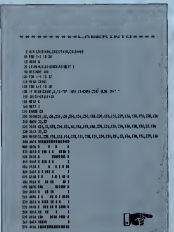
LABERINTO es un programa que nos envía HERMAN CORDOVA R. de VALPARAISO para computadora de bolsillo CASIO PB-700. La novedad de este programa es que tiene un subprograma que crea e inicializa las matrices necesarias, lo que se realiza una sola vez a lo largo de todo el juego. Como la PB tiene memoria continua las matrices creadas no se perderán a menos que se use el mando CLEAR. El programa se autoinicializa, por lo tanto es necesario que sea cargado con el mando CHAIN. En este caso sería: CHAIN "LABERINTO".

El juego consiste en recorrer un laberinto aprovechando al máximo el combustible del cual se dispone. La pantalla es sólo una ventana, que muestra una parte del laberinto, pero en el extremo derecho de ésta aparece un plano, completo de éste, con un punto intermitente que muestra la posición actual. Al mismo tiempo de avanzar el punto va borrando las líneas del plano lo que te indica cuánto te queda por recorrer. En el trayecto te encontrarás con estaciones de combustible (5), las cuales te permitirán llenar el estanque para así continuar en juego.

Los mandos son los siguientes:

IZQUIERDA	[4]
DERECHA	[6]
ARRIBA	[8]
ABAJO	[2]

Estamos seguros de que se van a divertir mucho, como también se van a instruir más con este juego tan dinámico que es "LABERINTO".





```
10 A=0:TM=42
20 CLS :PRINT L$40P1;
30 DRAW(112,0)-(120,0)-(120,1)-(112,1)
70 GOSUB 1700:GOSUB 1400:GOSUB 4000
90 I=0:L=1,P=1:S=0:T=10
100 REM          ******POINC-PAC*****
130 LOCATE 0,0:PRINT P$(I),P$(I+1),P$(I+2),P$(I+3)
135 IF POINT(8*I+1,8*P+8)=1 THEN LOCATE I,P:PRINT "O";:BEEP 1:GOSUB 3000
136 LOCATE I,P:PRINT "O";
138 IF POINT(145+I,1+P+9)=1 THEN S=S+10:IF S>=1500 THEN 1250
140 DRAW(143+I,1+P+9)
145 LOCATE 18,0:PRINT USING"##";T;:IF T>0 THEN T=T-1 ELSE 1200
148 DRAW(143+I,1+P+9)
150 GOTO 500+VAL(INKEY$)*10
160 REM**********
500 DRAW(143+I,1+P+9):T=T+.20:GOTO 145
510 GOTO 130
520 IF POINT(8*I,8*(P+1))=1 THEN BEEP :GOTO 140
521 IF P=2 THEN I=I+1 ELSE LOCATE I,P:PRINT " ";:P=P+1:GOTO 135
530 GOTO 130
540 IF POINT(8*(I-1),8*P)=1 THEN BEEP :GOTO 140
541 LOCATE I,P:PRINT " ";:I=I-1
550 GOTO 135
560 IF POINT(8*(I+1),8*P)=1 THEN BEEP :GOTO 140
561 LOCATE I,P:PRINT " ";:I=I+1
570 GOTO 135
580 IF POINT(8*I,8*(P-1))=1 THEN BEEP :GOTO 140
581 IF P=1 THEN I=I-1 ELSE LOCATE I,P:PRINT " ";:P=P-1:GOTO 130
590 GOTO 130
1200 LOCATE I,P:PRINT CHR$(240);:BEEP
1210 FOR J=1 TO 150:NEXT J:PRINT ,"PUNTOS=";S;
1220 ERASE P$:END
1250 R=R+5:BEEP :BEEP :BEEP 1:BEEP :BEEP 1:TM=30:GOTO 70
1400 FOR I=144 TO 159
1410 DRAW(I,10)-(I,25)
1420 NEXT I
1500 DRAWC(145,11)-(147,11):DRAWC(145,13)-(147,13):DRAWC(149,10)-(149,11)
1510 DRAWC(149,13)-(149,15)-(147,15):DRAWC(153,10):DRAWC(151,11)-(156,13)
1520 DRAWC(152,12)-(158,12)-(150,11)-(155,11):DRAWC(145,15)-(145,16)
1530 DRAWC(153,14):DRAWC(155,14):DRAWC(157,14):DRAWC(159,14)
1540 DRAWC(151,15)-(151,17)-(150,17):DRAWC(153,16):DRAWC(155,16)
1550 DRAWC(147,17)-(160,17):DRAWC(153,18)-(150,18)-(150,18)-(157,18)-(157,17)
1560 DRAWC(144,18)-(145,18):DRAWC(147,19)-(140,19)
1570 DRAWC(150,19)-(150,22)-(151,22)-(151,19)
1580 DRAWC(145,20)-(145,21)-(148,21)-(148,23)-(142,23)-(147,22)-(146,22)
1590 DRAWC(153,20)-(153,24)-(154,24):DRAWC(155,20)-(155,22)-(150,22)
1600 DRAWC(157,20)-(159,20):DRAWC(144,23)-(144,25)-(147,25):DRAWC(145,24)
1610 DRAWC(155,24)-(150,20)
1620 RETURN
1700 ERASE P$:DIM P$(17)*18
1710 FOR I=0 TO 17
1720 P$(I)=I$(I)
1730 NEXT I
1740 RETURN
2000 P$(I+P)=LEFT$(P$(I+P),I)+CHR$(A)+MID$(P$(I+P),I+2)
2020 RETURN
3000 A=32:I=INT(I)
3010 FOR I=I+1 TO 0:LOCATE 18,0:PRINT USING"##";I:NEXT I
3070 I=I-1:GOSUB 2000:RETURN
4000 P=0:A=230:DIM U(4)
4010 FOR Z=0 TO 2
4020 J=INT(4*RND)+Z*6+1:U(Z)=J:GOSUB 4100
4025 NEXT Z
4030 FOR Z=3 TO 4
4040 J=INT(4*RND)+3:FOR K=0 TO 4:IF J=U(K) THEN 4040 ELSE NEXT K
4050 U(Z)=J:GOSUB 4100
4060 NEXT Z:ERASE U:RETURN
4100 X=INT(18*RND)
4110 IF MID$(P$(J),X+1,1)=" " THEN IF X<>22 THEN GOSUB 2000:RETURN
4120 GOTO 4100
4200 REM LOBCAINTO,version II,Mar.1986
```









ATARI

Kuki

Belisario Eduardo Martinic Castro
Estudiante de Ingeniería Civil en Computación Universidad de Chile

En general, no es fácil realizar un programa BASIC Atari, que utilizando las facilidades de estas máquinas no excede de las 50 líneas. Es por esto último que me propuse realizar un juego que utilizase algo de sonido, algo de definición de caracteres, algo de manejo de pantalla, algo de animación y que además no superase las 50 líneas de largo. Sin embargo, mi esfuerzo fue inútil, y sólo conseguí dar a luz un juego, según la opinión de algunos, bastante entretenido, con algo de color, bastante de sonido, algo de animación, algo de definición de caracteres y de más de 100 líneas de largo (algo así como 7Kb de instrucciones). Este programa seguramente no ganará ningún premio por corrección y claridad, pero no por ello deja de merecer en algunos puntos, como por ejemplo, el sacrificio de comprensión y estructuración por rapidez de ejecución y ahorro de memoria, y la fácil modificación del programa general.

Kuki, nombre que recibe el juego, trata de las peripecias que debe vivir una voraz tortuga (Kuki) al enfrentar la necesidad vital de alimentarse. La trama se desarrolla en un campo de juego que contiene diversos elementos:

– Diversas frutas y verduras (manzanas, ajíes, plátanos, peras, uvas y coliflores)
– Calaveras o cruces
– Estrellas o pesos

y por el cual se debe llevar a Kuki (con el control joystick) de modo de satisfacer las necesidades alimenticias indicadas en la parte inferior del tablero. En ese lugar se muestra cada fruta con un número, éste es la cantidad necesaria de alimento que se debe comer. Cada vez que Kuki come una fruta o verdura, este número se decrementa y aumenta el puntaje (P en el tablero) en 10 puntos. Kuki sólo debe comer la cantidad indicada sin sobrealimentarse, ya que de lo contrario se le descontarán 20 puntos por alimento en exceso... pero, ¿cuál es la gracia de todo esto?

– Kuki debe realizar su tarea antes de que el tiempo límite acabe. El tiempo límite es indicado por el sonido de fondo, el cual aumenta de frecuencia con el tiempo.
– Kuki no debe tocar calavera alguna, ya que de lo contrario será aniquilada.
– Kuki no debe nunca quedar con puntaje bajo cero, ya que será aniquilada.
– Kuki puede saltar (un solo espacio a la vez) presionando el botón rojo del joystick y moviéndolo en la dirección deseada.
– Kuki puede comer puntos extras (pesos o estrellas), los cuales valen 100 puntos.
– Cada 2.000 puntos se obtiene una Kuki extra, si y sólo si, el número total de Kukis en reserva (parte superior del tablero), no supera las 4.
– Cada vez que Kuki termina con lo pedido, aumenta el nivel (N en el tablero), y con ello las dificultades... (las cuales, son un misterio).

Para correr el programa basta tipearlo y dar el comando RUN, ante lo cual la pantalla se oscurecerá durante algunos segundos y mostrará la presentación, la cual es autoinstructiva. Se recomienda tipear el programa con mucho cuidado, y antes de correrlo, guardar una copia en cinta, ya que un error en los datos podría ocasionar que el sistema se bloquee.

En caso de error es muy probable que éste no se presente en pantalla, ya que la instrucción

POKE SDMCTL, 0

hace que la pantalla se oscurezca para máxima velocidad. Se recomienda utilizar primero

POKE SDMCTL, 34

con lo cual los posibles errores aparecerán en pantalla.

Para el juego se han utilizado las siguientes constantes y variables: M

RAMTOP	Dirección que contiene la ubicación de memoria más alta usada por BASIC
ICH	Dirección que contiene la base interna de caracteres
CHBASE	Dirección que contiene la base definida de caracteres
SDMCTL	Registro sombra del control de pantalla
AUDC1 ... AUDC4	Registros de audio
AUDF1 ... AUDF4	Registros de frecuencia de audio
AUDCTL	Registro de control de audio
COL0 ... COL4	Registros de color
PT	Base del puntaje extra.
PTM	Paso para el puntaje extra
M	Máximo de objetos por nivel
CF	Color para el fondo (Pasto)
XMIN	Límite vertical izquierdo del campo de juego
XMAX	Límite vertical derecho del campo de juego
YMIN	Límite horizontal inferior del campo de juego
YMAX	Límite horizontal superior del campo de juego
CL	Color de la muralla.
CAMBIOS	Número de caracteres cambiados
CT	Color de Kuki
F (7)	Arreglos que contiene la cantidad total de calaveras (cruces) frutas y pesos (estrellas)

CF (7)	Arreglo que contiene el color asociado al arreglo F
NF (7)	Arreglo que contiene el numero de elementos de F presentados en el juego
C1 $, C2 $, C3 $	Strings utilizados para la rutina de transformación Hexadecimal-Decimal
M$	String que contiene la rutina de máquina para el traspaso de caracteres desde ROM a RAM
F, I	Contadores de ciclos
C, D	Datos temporales varios
B1, B2, BASE	Bases para los set de caracteres (definidos e internos)
V	Valor decimal o posición en el tablero
Q	Resultado de la rutina de máquina (Descartable)
N	Contador vario
NIVEL	Nivel de dificultad del juego
P	Puntos
UT	Ultimo color (forma) de Kuki
XT	Posición de las Kukis restantes en el tablero
X, Y	Posición de Kuki en el campo de juego
FR	Frecuencia del sonido de fondo
ITERA	Número de iteraciones para el sonido de fondo
VL, VH	Valor alto y bajo de un valor decimal al pasar a hexadecimal
XF,YF	Posiciones aleatorias de los objetos en el campo de juego
S	Lectura del joystick 1 (Modificada)
B	Lectura del trigger 1 (Modificada)
C2, C4	Banderas para el nivel de dificultad Las cuales estructuran al programa en la siguiente forma

0	Instrucción para la deshabilitación del teclado
10-70	Definición de constantes y variables
100-220	Rutinas de definición y traspaso del juego de caracteres, y asignación del modo gráfico
240-340	Presentación y comienzo del juego
500-570	Asignación y determinación de las constantes de juego
580-590	Ciclos de iteración sobre la rutina de sonido
600-950	Programa principal
960-990	Rutina de muerte de Kuki
1000	Datos para la rutina de traspaso de caracteres
1010-1200	Datos para los juegos definidos de caracteres
2000-2020	Rutina de transformación desde notación Hexadecimal a valor decimal alto-bajo
2100-2120	Rutina para determinar un espacio libre en el campo de juego
2200-2220	Rutina de sonido para indicar comida de fruta
2300-2320	Rutina de sonido para indicar comida en exceso
2400-2440	Rutina de sonido para el cambio de nivel
2500-2520	Rutina de sonido para indicar comida de puntos extras
2600-2640	Rutina para crear Kuki extra
2700	Rutina para generar una ubicación aleatoria en el campo de juego

```
0 POKE 621,1
10 READ RAMTOP,ICH,CHBASE,SDMCTL,AUDC1,AUDC2,AUDC3,AUDC4,AUDF1,AUDF2,AUDF3,AUDF4
,AUDCTL,COL0,COL1,COL2,COL3,COL4
20 DATA 106,57344,756,559,53761,53763,53765,53767,53760,53762,53764,53766,53768,
708,709,710,711,712
30 READ N,CF,XMIN,XMAX,YMIN,YMAX,CL,CAMBIOS,CT,PTH,PM
40 DATA 15,31,1,18,3,19,142,15,129,2000,2000
50 DIM F(7),CF(7),NF(7),C$(64),C1$(16),C2$(16),M$(32)
60 FOR I=0 TO 7:READ D:CF(I)=D:NEXT I
70 DATA 141,166,167,40,41,138,43,140
100 POKE SDMCTL,0
110 PAG=PEEK(RAMTOP)-9
120 B1=(PAG+1)*256:B2=(PAG+5)*256
130 POKE RAMTOP,PAG
140 GRAPHICS 17:POKE COL0,40:POKE COL1,230:POKE COL2,66:POKE COL3,14:POKE COL4,0
150 RESTORE 1000:READ C$
160 FOR I=1 TO LEN(C$) STEP 2:GOSUB 2000:M$((I+1)/2,(I+1)/2)=CHR$(V):NEXT I
170 Q=USR(ADR(M$),B1,ICH):Q=USR(ADR(M$),B2,ICH)
180 FOR N=1 TO CAMBIOS
190 READ C,C1$,C2$
200 FOR I=1 TO LEN(C1$) STEP 2:C$=C1$:GOSUB 2000:POKE B1+C*8+(I-1)/2,V:C$=C2$:GO
SUB 2000:POKE B2+C*8+(I-1)/2,V:NEXT I
```

```
210 NEXT N
220 POKE SDMCTL,254
240 COLOR 0:FOR I=0 TO 19:PLOT I,0:DRAWTO I,23:NEXT I
250 COLOR 3:PLOT 0,0:DRAWTO 0,14:DRAWTO 19,14:DRAWTO 19,0:DRAWTO 0,0:PLOT 1,10:D
RAWTO 18,10
260 POSITION 5,2:? #6;"MICROBYTE":POSITION 9,4:? #6;"Y"
270 POSITION 5,6:? #6;"D E M A C".POSITION 4,8:? #6;"INFORMATICA":POSITION 5,12:
? #6;"PRESENTAN"
280 COLOR 75:PLOT 2,16:DRAWTO 2,20:PLOT 5,16:DRAWTO 3,18:DRAWTO 5,20:PLOT 12,16:
DRAWTO 12,20:PLOT 15,16
290 DRAWTO 13,18:DRAWTO 15,20
300 COLOR 85:PLOT 7,16:DRAWTO 7,20:DRAWTO 10,20:DRAWTO 10,16
310 COLOR 73:PLOT 17,16:DRAWTO 17,20
320 POSITION 4,23:? #6;"pulse  start"
330 IF PEEK(53279)<>6 THEN 310
340 NIVEL=1:P=0:UT=129:XT=13
400 POSITION 2,1:? #6;"N:";NIVEL;" P:";P
410 COLOR 129:PLOT 13,1:DRAWTO 17,1
420 COLOR CL
430 PLOT 0,0:DRAWTO 0,23:DRAWTO 19,23:DRAWTO 19,0:DRAWTO 0,0
440 PLOT 1,2:DRAWTO 18,2:PLOT 1,20:DRAWTO 18,20
450 POKE CHBASE,B1/256
500 F(0)=30+12*NIVEL
510 FOR I=1 TO 7:F(I)=INT(RND(0)*M)+1:NF(I)=INT(RND(0)*F(I))+1:NEXT I
520 COLOR CF
530 FOR Y=YMIN TO YMAX:PLOT XMIN,Y:DRAWTO XMAX,Y:NEXT Y
540 FOR I=0 TO 7:COLOR CF(I):FOR N=1 TO F(I):GOSUB 2100:NEXT N:NEXT I
550 FOR X=3 TO 13 STEP 5:FOR Y=21 TO 22
560 COLOR CF((X+2+15*(Y-21))/5):PLOT X,Y:? #6;":";NF((X+2+15*(Y-21))/5)
570 NEXT Y:NEXT X
580 SOUND 0,0,0,0:POKE AUDCTL,4:POKE AUDC1,168:POKE AUDC2,168
590 COLOR CT:GOSUB 2100:BASE=B2/256
600 FOR FR=254 TO 80 STEP -2:FOR ITERA=1 TO 5-NIVEL
610 POKE AUDF1,FR:POKE AUDF3,FR/2
620 BASE=B1/256*(BASE=B2/256)+B2/256*(BASE=B1/256):POKE CHBASE,BASE
630 R=STICK(0):B=STRIG(0):POKE 77,0
640 S=(R=15)+2*(R=14)+3*(R=7)+4*(R=13)+5*(R=11)+128:IF S=128 THEN S=129
650 IF S<>129 THEN COLOR CF:PLOT X,Y:UT=CT*(UT<>CT)+S*(UT=CT)
680 IF  NOT B THEN 720
690 UX=X:X=X+(S=131)-(S=133):X=XMIN*(X<XMIN)+X*(XMIN<=X AND X<=XMAX)+XMAX*(X>XMA
X)
700 UY=Y:Y=Y+(S=132)-(S=130):Y=YMIN*(Y<YMIN)+Y*(YMIN<=T AND Y<=YMAX)+YMAX*(Y>YMA
X)
710 GOTO 740
720 UX=X:X=X+2*(S=131)-2*(S=133):X=XMAX*(X<XMIN)+X*(XMIN<=X AND X<=XMAX)+XMIN*(X
>XMAX)
730 UY=Y:Y=Y+2*(S=132)-2*(S=130):Y=YMAX*(Y<YMIN)+Y*(YMIN<=Y AND Y<=YMAX)+YNIN*(Y
>YMAX)
740 IF NIVEL=1 THEN 760
750 C2=C2+1:IF C2=10 THEN C2=0:GOSUB 2700:COLOR CF(INT(RND(0)*8)):PLOT XF,YF
760 IF NIVEL<>3 THEN 780
770 C3=C3+1:IF C3=30 THEN C3=0:GOSUB 2700:COLOR CL:PLOT XF,YF
780 IF NIVEL<4 THEN 800
790 C4=C4+1:IF C4=40 THEN C4=0:COLOR CL:GOSUB 2700:PLOT XF,YNIN:DRAWTO XF,YMAX
800 LOCATE X,Y,V
810 IF V=CF(0) THEN GOTO 960
815 IF V=CL THEN X=UX:Y=UY:GOTO 860
820 IF V=CF OR V=CT OR V=130 OR V=131 OR V=132 OR V=133 THEN 860
830 IF V=CF(7) THEN P=P+100:GOSUB 2500:GOTO 860
840 FOR F=1 TO 7:IF V=CF(F) THEN NF(F)=NF(F)-1:P=P+10:POP :GOSUB 2200:GOTO 850
845 NEXT F
```

```
850 IF NF(F)<0 THEN NF(F)=0:P=P-30:GOSUB 2300:IF P<0 THEN P=0:GOTO 960
860 COLOR VT:PLOT X,Y
865 IF P>PM THEN PM=PM+PTM:IF XT>13 THEN GOSUB 2600
870 POSITION 4,1:? #6;NIVEL:POSITION 8,1:? #6;P;" "
880 POSITION 5,21:? #6;NF(1);" ":POSITION 10,21:? #6;NF(2);" ":POSITION 15,21:?
#6;NF(3);" "
890 POSITION 5,22:? #6;NF(4);" ":POSITION 10,22:? #6;NF(5);" ":POSITION 15,22:?
#6;NF(6);" "
900 FOR I=1 TO 6:IF NF(I) THEN POP :GOTO 950
910 NEXT I
920 GOTO 2400
950 NEXT ITERA:NEXT PE
960 POKE AUDC2,142:POKE AUDC4,142:FOR I=0 TO 255:POKE AUDF2,I:POKE AUDF4,255-I:P
OKE COL4,255-I:NEXT I
970 POKE AUDC2,0:POKE AUDF2,0:POKE AUDC4,0:POKE AUDF4,0
980 XMIN=1:COLOR CF:PLOT X,Y:COLOR 0:PLOT XT,1:XT=XT+1:IF XT<>18 THEN 600
990 POSITION 2,20:? #6;"JUEGO  TERMINADO":POSITION 4,23:? #6;"PULSE OPTION":IF P
EEK(53279)<>3 THEN 990
999 GOTO 240
1000 DATA 8686587D86586D86585D86584D2A400A00184D196D8C0D9F6E5D6E7DAC0D0F06
1010 DATA 1,000083C7C7830000,000083C7C7830000
1020 DATA 2,454583C7C7834444,454583C7C7834444
1030 DATA 3,0000BDC3F3C3BD00,0000BDC3F3C3BD00
1040 DATA 4,444483C7C7834545,444483C7C7834545
1050 DATA 5,0000BDC3CFC3BD00,0000BDC3CFC3BD00
1060 DATA 6,058F07078383C1C0,41E3E38707070604
1070 DATA 7,030177F7F7F7F7E3,0018183C3C7EE7C3
1080 DATA 8,0383C1E0E0E0C183,C0C1830707078 3C1
1090 DATA 9,8180C1E3E7EFCF8F,810183C7CFEFFFF7
1100 DATA 10,81C3E7C381818100,00818181C3E7C381
1110 DATA 11,020301AA55A24180,40C08055AA450501
1120 DATA 12,81E785E7A1E78100,000102C783824400
1130 DATA 13,C3A5E7C3997EC37E,3C66C38181C3663C
1140 DATA 31,0082C7E3C7834100,0082C7E3C7834100
1200 DATA 14,FF00FF00FF00FF00,FF00FF00FF00FF00
2000 VL=ASC(C$(I,I))-ASC("0"):IF VL>9 THEN VL=VL-7
2010 VH=ASC(C$(I+1,I+1))-ASC("0"):IF VH>9 THEN VH=VH-7
2020 V=VL+VH*16:RETURN
2100 X=INT(RND(0)*(XMAX-XMIN+1))+XMIN:Y=INT(RND(0)*(YMAX-YMIN+1))+YMIN
2110 LOCATE X,Y,V:IF V<>CF THEN 2100
2120 PLOT X,Y:RETURN
2200 POKE AUDC2,170
2210 FOR I=255 TO 40 STEP -10:POKE AUDF2,I:NEXT I
2220 POKE AUDC2,0:POKE AUDF2,0:RETURN
2300 POKE AUDC2,142
2310 FOR I=40 TO 255 STEP 10:POKE AUDF2,I:NEXT I
2320 POKE AUDC2,0:POKE AUDF2,0:RETURN
2400 XMIN=1:NIVEL=NIVEL+1:IF NIVEL>16 THEN NIVEL=0
2410 POKE AUDC2,170:POKE AUDC4,170
2420 FOR I=255 TO 0 STEP -1:POKE AUDF2,I:POKE AUDF4,255-I:POKE COL3,I:NEXT I
2430 POKE AUDC2,0:POKE AUDC4,0:POKE AUDF2,0:POKE AUDF4,0:POKE COL3,14+16*(NIVEL-
1)
2440 COLOR 0:PLOT 1,21:DRAWTO 18,21:PLOT 1,22:DRAWTO 18,22:GOTO 500
2500 POKE AUDC2,170
2510 FOR I=1 TO 10:POKE AUDF2,60:FOR T=1 TO 5:NEXT T:POKE AUDF2,0:NEXT I
2520 POKE AUDF2,0:RETURN
2600 XT=XT-1:FOR N=1 TO 5:COLOR 0:PLOT XT,1:GOSUB 2500:COLOR CT:PLOT XT,1:NEXT N
:RETURN
2700 XF=INT(RND(0)*(XMAX-XMIN+1))+XMIN:YF=INT(RND(0)*(YMAX-YMIN+1))+YMIN:RETURN
```

Uso de Sprite en Commodore

Uno de los mayores problemas que existen dentro de los usuarios de COMMODORE 64 es la creación de un Sprite, ya que es muy engorroso calcular las famosas Datas y luego agregarlas a un programa. Este programa le permitirá a los usuarios dibujar su Sprite y luego, que el computador le dé las datas automáticamente. Ud. tendrá 8 Sprites disponibles para crear, sin tener que borrar el anterior. Una vez que se le hayan acabado los 8 Sprites disponibles, deberá comenzar de nuevo el programa.

Para usar este programa deberá correrlo, con lo que aparecerá la pantalla cubierta de puntos, en la cual podrá dibujar con asteriscos la figura que usted desee. Luego mueva el cursor hasta que esté en la línea que dice "dibujo" y presione RETURN. Esto le mostrará su dibujo en uno de los sprite que estarán a la derecha de su pantalla. Si desea ver los datos, ponga el cursor sobre la línea "datos" y presione RETURN. Esto le mostrará los datas en pantalla. Si desea corregir un asterisco vaya a "dibujo" y cámbielo por un punto. Esto hágalo hasta que quede conforme, ya que al ir a la rutina "datos" no podrá volver a corregir su sprite, ya que después de pasar a la rutina "datos" pasa automáticamente al próximo sprite. Esperamos que este programa le ayude a entender un poco más el funcionamiento de los "Sprite" ■

```
10 REM SPRITES
20 SN=0
30 FORF=12288TO12798:POKEF,255 NEXTF
40 PRINTCHR$(147),
50 POKE53280,0:POKE53281,0
60 FORT=1TO21 PRINT"■........................" NEXT
70 PRINT"GOTO200:REM■ DIBUJO ■" PRINT"GOTO300:REM■ DATOS  ■"
80 PRINT"[illegible]"
90 V=53248 POKEV+21,255 FORF=0TO7:POKE(V+39+F),1 NEXT
100 FORF=0TO6STEP2:POKEV+F,250:POKEV+F+8,43:NEXT:POKEV+16,240
110 POKEV+1,50 POKEV+3,98:POKEV+5,146:POKEV+7,194
120 POKEV+9,50 POKEV+11,98:POKEV+13,146:POKEV+15,194
130 FORF=0TO7:POKE(2040+F),192+F:NEXT
140 END
200 C=8:G=0
210 FORY=0TO20
220 FORX=1TO24
230 P=PEEK(1023+X+40*Y)
240 C=C-1:IFP=42THENQ=Q+2↑C
250 AD=SN*64+12288+G
260 IFC=0THENC=8:G=G+1:POKEAD,Q:Q=0
270 NEXTX
280 NEXTY
290 END
300 PRINTCHR$(147):PRINT
310 FORT=0TO62
320 PRINTPEEK(SN*64+12288+T),
330 K=K+1:IFK=3THENK=0:PRINT
340 NEXT
350 SN=SN+1
355 PRINT "    ■OPRIMA UNA TECLA PARA CONTINUAR■"
360 GETA$:IFA$=""THENGOTO360
380 GOTO40
390 END
```

trarse en estructurar los bloques que no se expresan en la base elegida

Una vez finalizado el Parsing, se estructura el diagrama segun lo visto anteriormente

Veamos ahora otro ejemplo donde aplicamos el Parsing y luego el teorema de la estructuración

– Fig A Diagrama original y primera extracción de programas primos

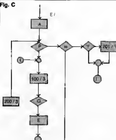
Fig. A

NRO DE BLOQUE — CANT DE NODOS

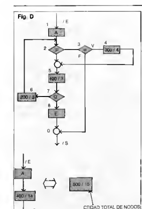
– Fig B Se reemplazan los programas primos de la Fig A por los bloques correspondientes y se realiza una nueva extracción

Fig. B

– Fig C Se reemplazan los programas primos de la Fig B por los bloques correspondientes y se realiza una nueva extracción



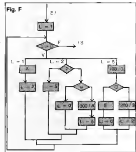
– Fig D Se reemplazan los programas primos de la Fig C por los bloques correspondientes y se llega a un programa no estructurado el cual puede resumirse en un unico bloque (Bloque Nº 500/15)



– Fig E Resulta de aplicar el teorema de la estructuracion al diagrama obtenido en la Fig D

Fig. E

– Fig F Eliminación de ramas Se obtiene un diagrama estructurado Cuando este diagrama obtenido se codifique en algun lenguaje (que admita programas estructurados) cada bloque será reemplazado por el conjunto de instrucciones que corresponda



Conclusiones

Podemos decir que la programación estructurada es una buena herramienta para las técnicas de programación pues agiliza el seguimiento de un programa ya sea para extraer su función o detectar errores

Por otro lado presenta el inconveniente de ocupar más espacio en memoria, puesto que seguramente habrá que repetir determinadas rutinas lo que puede evitarse trabajando con programas no estructurados

El programador decidirá, en su momento qué es mejor

– Más espacio, menos claridad o
– Menos espacio más claridad

Bibliografía:
– Structured Programming Theory and Practice Linger Mills and Witt - Ed Adison Wesley
– Curso de Electrónica e Informática Ed Hyspamérica, 1986



Marcela A Roma tiene 22 años Cursa actualmente 4° año de Licenciatura en Informática en la Universidad Nacional de La Plata, Facultad de Ciencias Exactas habiendo finalizado de cursar el año pasado las materias correspondientes a Analista en Computación

Un utilitario en Turbo Pascal le permite integrar las capacidades de Dbase III y Word Perfect

TRASPASO DE ARCHIVOS DBASE A WORD PERFECT

Pablo Bañados Norero

El paquete de procesamiento de textos Word Perfect es extremadamente util en la mayoria de las situaciones Entre las posibilidades del programa está la facilidad de crear una lista de registros cuyos campos pueden ser luego extraidos individualmente por un texto permitiendo de esta manera por ejemplo llenar una carta formulario con los datos individuales de cada destinatario

Lamentablemente el formato utilizado por el archivo auxiliar conteniendo los registros a fusionar, es especifico de este programa no permitiendo el uso de datos generados a través de bases de datos como la ampliamente difundida dBase III

El programa que se presenta a continuación efectúa la transformación del formato de un archivo dBase III para ser usado por Word Perfect quedando estos ultimos grabados en un nuevo archivo Se ofrece también la alternativa de trasladar sólo los campos que le interesan permitiendo de esta manera reducir el tamaño del archivo destino

Para esto fue necesario determinar la estructura de los archivos de uno y otro programa

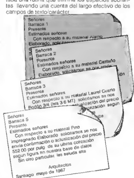
Word Perfect tiene una estructura sumamente sencilla Cada campo del registro termina con una serie de dos caracteres cuyos valores ascii son 18 y 10 respectivamente A su vez cada registro termina con otros dos caracteres, valores ascii 5 y 10 Obviamente se requiere que todos los registros tengan la misma cantidad de campos

Dbase III tiene en cambio una estructura mas compleja Los primeros 32 caracteres presentan la información general del archivo De éstos los caracteres numeros 5 y 6 contienen el numero total de registros del archivo siendo el primero el byte menos significativo Al respecto ver procedimiento ' Global en donde el largo total queda registrado en la variable ' total

Posteriormente se presenta la información general de cada uno de los campos en grupos de 32 bytes Los diez primeros contienen el nombre del campo el numero doce contiene el tipo (carácter numérico lógico etc) y el 17 contiene el numero máximo de caracteres del campo Para terminar la definicion de campos dBase graba un ascii 13 (CR) como primer carácter del siguiente campo seguido de un ascii 0 La lectura de esta información se efectua en el procedimiento estructura ' quedando los datos incluidos en los arrays nombre tipo y largo Al respecto se hace notar que el programa acepta un máximo de 32 campos, que es la cantidad que despliega normalmente dBase III al momento de definir/modificar la estructura de una base de datos Si se están usando más campos basta con redefinir la constante max-campo aunque se hace necesario también la presentacion de la información en pantalla (procedimiento informa)

En seguida viene la información de cada registro propiamente tal en que cada campo ocupa el espacio requerido por su numero máximo de caracteres Además de esto los registros van antecedidos por otro byte (conteniendo normalmente un ascii 32)

Cabe hacer notar que en los campos de carácter/texto dBase rellena con ascii 32 (espacios en blanco) los bytes no ocupados lo que para efectos de los objetivos del programa que se presenta tendria desagradables consecuencias ya que introduciria espacios en blanco que desformatearian el texto Por esto el programa tiene implementada una rutina que elimina los espacios sobrantes llevando una cuenta del largo efectivo de los campos de texto/carácter



El programa fue escrito en un computador IBM y en lenguaje Turbo Pascal. El tratamiento en la lectura y escritura de archivos no es de lo más eficiente, ya que lo hace carácter por carácter, sin embargo esto hace al programa más flexible y fácil de entender; la aplicación no requiere además mayor optimización en este punto.

Los usuarios más experimentados en dBase III notarán que este programa es muy similar a la orden "copy to", con la opción "sdf". Sin embargo, en ella no es posible dejar el archivo ".txt" resultante listo para el uso que se requiere, ya que a) resulta muy difícil incluir los códigos delimitadores requeridos, b) introduce un carácter de retorno (ascii 13 y ascii 10) al final de cada registro y c) no elimina los espacios en blanco a que hacíamos mención más arriba. Sin embargo, el mayor interés de lo que presentamos es que una vez comprendida cabalmente la estructura de los archivos dBase III, estos pueden ser usados para construir la aplicación que el programador requiera y no sólo para la implementación presentada aquí.

Si usted tiene el lenguaje Turbo Pascal o no quiere tipear el programa, le puedo enviar copia del archivo, fuente y compilado. Escribir o llamar a

Pablo Bañados N.

Diego de Deza 1415, Santiago (10), Chile
Teléfono 2204384

Pablo Bañados Norero se recibió de Arquitecto en la Universidad de Chile en 1985. Desde 1986 trabaja como profesional de libre ejercicio en su oficina particular, desarrollando proyectos para oficinas públicas como Compañía de Teléfonos, así como para particulares.
Autodidacta de la computación, ha realizado cursos de sistematización de conocimientos auspiciados por IBM. Pasando por una variedad de computadores, ha publicado programas para ellos anteriormente en Microbyte (Topo 3d, No +!).
En la actualidad cuenta con un computador IBM de intenso uso en su oficina, en el cual ha desarrollado algunas asesorías profesionales, investigando también las posibilidades de difusión de la informática en su profesión.

Tabla 1

```
. list all fields clase, cod, material, uni
Record# CLASE  COD MATERIAL            UNIDAD   PRECIO DSCTO PROVEEDOR
      1 CB     103 ALAMO ELABORADO     pulg     540.00     0 BARRALA 1
      2 CB     550 ALERCE ELABORADO    pulg    3600.00     0 BARRALA 1
      3 CB    4585 CASTAÑO ELABORADO   pulg    2400.00     0 BARRALA 2
      4 CB    5046 CIPRES ELABORADO    pulg    1000.00     0 BARRACA 1
```

```
program dbase wp;
(modifica archivos de Dbase III para uso de listas en Word Perfect
 autor :  Pablo Bañados Norero
 fecha :  Mayo 1987)

const
max campo = 32;

type
palabra            =string[10];
frase              =string[255];

var
a                  :char;
w,archivo          :frase;
filvar,destino     :file of char;
l,total,campos     :integer;
campo              :array [1..max campo] of frase;
nombre             :array [1..max campo] of palabra;
tipo               :array [1..max campo] of char;
largo,largoreal    :array [1..max campo] of integer;
dato               :array[1..32] of char;
conf               :array [1..max campo] of boolean;
function tecla:boolean;
var
ch :char;
begin
repeat
   read (Kbd,Ch)
until Upcase(Ch) in ['Y', 'N', #27];
if Upcase(Ch) in ['N', #27] then tecla:=false
else tecla:=true;
end;

procedure confirma;
begin
conf[l]:=tecla;
if conf[l] then writeln('SI')
else writeln('NO');
end;

procedure global;
var
l   :integer;
a,b :char;

begin
for l:=1 to 5 do
    read(filvar,a);
read(filvar,b);
total:=ord(b)*256 +ord(a);
for l:=7 to 32 do
    read(filvar,a);
end;

procedure estructura;
label continua;
var
ct :integer;
begin
ct:=1;
repeat
      read(filvar,a);
      if ord(a)=13 then goto continua;
      for l:=2 to 32 do
          read(filvar,dato[l]);
      nombre[ct]:=a;
      for l:=2 to 10 do
          nombre[ct]:=nombre[ct]+dato[l];
      tipo[ct]:=dato[12];
      largo[ct]:=ord(dato[17]);
      ct:=ct+1;
continua : until ord(a)=13;
campos:=ct-1;
```

Continua pág. 38

```
readfilvar,a);read(filvar,a);
end;

procedure informa;

var
ch        :char;
despliegue :integer;
begin
clrscr;
gotoxy(1,1);
write('ARCHIVO FUENTE ');
gotoxy(20,1);
writeln(' = ',archivo);
write('ARCHIVO DE DESTINO ');
gotoxy(50,24);write('PROCESANDO ARCHIVO =');
gotoxy(20,2);
writeln(' = ',w);
gotoxy(50,1);
writeln('TOTAL DE REGISTROS = ',total);
gotoxy(4,4);
write('CAMPO     TIPO    PORTE DEC CONF ');
gotoxy(44,4);
write('CAMPO     TIPO    PORTE DEC CONF ');
gotoxy(4,5);
for l:=1 to 32 do write('=');
gotoxy(44,5);
for l:=1 to 32 do write('=');
if campos<=16 then despliegue:=campos
   else despliegue:=16;
gotoxy(1,6);
for l:=1 to despliegue do
    begin
    gotoxy(1,wherey);
    write(l);
    gotoxy(4,wherey);write(nombre[l]);
    gotoxy(15,wherey); write(tipo[l]);
    gotoxy(22,wherey); writeln(largo[l]);
    end;
if campos>despliegue then
   begin
   gotoxy(40,6);
   for l:=despliegue+1 to campos do
        begin
        gotoxy(41,wherey);
        write(l);
        gotoxy(44,wherey);write(nombre[l]);
        gotoxy(55,wherey); write(tipo[l]);
        gotoxy(62,wherey); writeln(largo[l]);
        end;
   end;
gotoxy(1,24);write('TRANSLADA TODOS LOS CAMPOS (Y/N)?');
  if tecla=false then
     begin
     gotoxy(1,24);write('                                        ');
     gotoxy(1,24);write('INGRESE Y/N EN COLUMNA CONF ');
     gotoxy(31,6);
     for l:=1 to despliegue do
         begin
         gotoxy(32,wherey);
         write(' ');
         gotoxy(32,wherey);
         confirma;
         end;
     if campos>despliegue then
        begin
        gotoxy(71,6);
        for l:=despliegue+1 to campos do
            begin
            gotoxy(72,wherey);
            write(' ');
            gotoxy(72,wherey);
            confirma;
            end;
        end;
     end
  else for l:=1 to campos do conf[l]:=true;
end;

procedure lee_escribe;
label continua;
var
l,m,n   :integer;

procedure cola(x :integer);
begin
a:=chr(x);
write(destino,a);
end;

begin
for l:=1 to total do
    begin
    gotoxy(70,24);write(l);
    for m:=1 to campos do
        begin
        campo[m]:='';
        for n:=1 to largo[m] do
            begin
            read(filvar,a);
            campo[m]:=campo[m]+a;
            end;
        if conf[m]=true then
           begin
           if (tipo[m]='C') then begin
                                 largoreal[m]:=largo[m];
                                 for n:=largo[m] downto 1 do
    begin
    if ord(campo[m][n])<33 then largoreal[m]:=largoreal[m]-1
       else goto continua;
    end;
end
           else largoreal[m]:=largo[m];
continua : for n:=1 to largoreal[m] do
               begin
               a:=campo[m][n];
               write(destino,a);
               end;
           cola(13);
           cola(10);
           end;
        end;
        cola(5);
        cola(10);
        read(filvar,a);
    end;
close(filvar);
close(destino);
end;

begin
for l:=1 to max_campo do campo[l]:='';
clrscr;
writeln('Ingrese archivo fuente, con drive, sin extension ');
readln(con,archivo);
w       :=archivo + '.wp';
archivo:=archivo + '.dbf';
assign(filvar,archivo);
assign(destino,w);
rewrite(destino);
reset(filvar);
global;
estructura;
informa;
gotoxy(1,24);write('                                        ');
gotoxy(1,24);write('ESPERE UN MOMENTO ');
lee_escribe;
end.

^Z
```

Comparta las experiencias, resuelva sus dudas y participe con nosotros en esta sección orientada a los PC y compatibles.

Dr. PC

Hector A Miranda Riquelme

En este numero iniciamos esta nueva sección que pretende establecer una comunicación activa entre MICROBYTE y sus lectores Queremos que usted estimado lector y amante de los computadores personales nos escriba participándonos sus inquietudes sus dudas sus problemas sus trucos y sus descubrimientos en la utilización cotidiana de su IBM-PC o compatible

Todo lo que debe usted hacer es escribirnos contándonos su duda o descubrimiento Nosotros responderemos a través de esta sección, permitiendo a nuestros lectores conocer un poco más de esta máquina tan asombrosa como es el PC

Problemas del cambio de diskettes

Una inquietud que nos ha sido formulada muchas veces y que en más de una oportunidad hemos visto traducida en un gran problema

P. Me encontraba trabajando en dBASE III, agregando registros a un archivo En un momento dado quise seguir egregando dichos registros en otro diskette, donde tenia un respaldo de mi archivo original, por lo que abrí el drive, saqué el primer diskette introduje el segundo diskette, cerré el drive y continué con el APPEND que había interrumpido Al finalizar el proceso, salí de dBASE III, y al examinar con un comando DIR de qué tamaño me había quedado el archivo, me encontré con que en mi segundo diskette aparecían archivos que yo jamás había grabado en él y, más aun, comprobé posteriormente que algunos de esos archivos eran programas " COM" que al ser llamados no funcionaron, sino que dejaron "colgando' al computador Para mayor desgracia mía, otros archivos que yo tenía grabados en ese diskette "desaparecieron" ¿Qué es lo que le puede haber sucedido a este diskette?

R. Nos encontramos aquí ante uno de los graves inconvenientes del PC-DOS Es un inconveniente, porque es un problema no documentado y porque no ofrece un mensaje de alerta al usuario

Cuando se usa un diskette en el PC, su directorio y su FAT son almacenados en memoria para posibilitar una mayor eficiencia en la lectura y grabación de información Esta región de memoria es actualizada cada vez que se ejecuta una operación sobre el diskette Cuando se efectua una operación de lectura el PC lee nuevamente los dos elementos desde el diskette y los guarda en la memoria Cuando se efectua una operación de escritura o grabacion el PC prepara" lo que va a grabar hace las correspondientes modificaciones al directorio y a la FAT en la memoria y a continuación graba sobre el diskette ¿Qué es lo que graba? La información y una copia del directorio y de la FAT que tiene en memoria

Ahora bien ¿qué es lo que pasa cuando se cambia el diskette? Al cambiar un diskette el computador no es capaz de darse cuenta de lo que ha sucedido y para todos sus efectos, considera que el directorio y la FAT que guarda en su memoria son los vigentes hasta que se realice una nueva operación sobre el diskette

Supongamos que como en el caso planteado estamos trabajando con un archivo dBASE agregando registros El dBASE conserva un buffer de memoria en el cual va guardando temporalmente los datos que se le van ingresando para llevarlos al disco periódicamente En dicho proceso de grabación, el DOS actualiza el directorio y la FAT en el diskette Ahora si estamos agregando registros como en el caso planteado y de pronto cambiamos el diskette el computador no se da cuenta de que hemos hecho esto y sigue trabajando con el mismo directorio y la misma FAT del disco anterior Por eso al actualizar el dBASE los datos en el archivo del nuevo diskette graba también sobre él la FAT y el directorio antiguos ocasionando problemas como los descritos y otros más En realidad, lo que pasa es que lo que habia antes en el diskette sigue estando físicamente grabado allı pero no se puede accesar porque para el DOS sencillamente ya no existe

¿Cómo podemos evitar este accidente? En este aspecto el pequeño CP/M era más precavido y no dejaba ocupar un nuevo diskette sin antes haber ejecutado el famoso Control C Pero ya vemos que el PC DOS no es así Por lo tanto debemos de alguna forma avisarle al sistema operativo que le hemos cambiado el disco En el caso del dBASE lo mejor es cerrar el archivo en el primer disco, con un comando USE en seguida cambiar el diskette y abrir el nuevo archivo con un comando USE ARCHIVO y continuar el APPEND en el nuevo diskette En el caso de otros programas de aplicación primero haga que el computador se entere de que usted le ha cambiado el disco mediante un simple comando de lectura del directorio Con esto lo obliga a obtener una nueva copia en memoria de la FAT y del directorio

Un programa en Lotus para resolver uno de los problemas clásicos en computacion.

ANALISIS DE COLAS CON LOTUS 1-2-3

Guillermo Beuchat S.



Existen muchas situaciones de la vida real que pueden describirse mediante la llamada "teoría de colas" tales como los clientes que esperan en fila ante la ventanilla de un banco, los aviones que esperan aterrizar en un aeropuerto o las llamadas que llegan hasta una central telefonica. La característica comun a todas estas situaciones es que la capacidad de servicio o atención es limitada, por lo que algunas de las unidades o personas que llegan deberan formar una cola y esperar su turno.

En un numero anterior de Microbyte (1) hemos tratado este tema mediante un enfoque de simulación basado en el uso de numeros aleatorios para generar tiempos entre llegadas sucesivas a las unidades de servicio. Sin embargo existe un enfoque analítico que permite llegar a soluciones adecuadas cuando se dan ciertas condiciones bastante comunes en situaciones reales. En este trabajo se presenta un modelo de análisis de sistemas de espera desarrollado en LOTUS 1 - 2 - 3, que permite realizar facilmente un análisis de sensibilidad sobre los diversos parámetros que describen el sistema.

Caracteristicas de las colas

A fin de plantear el problema desde un punto de vista analítico es necesario considerar básicamente tres aspectos

a) **Las llegadas.** Las llegadas del sistema se describen por su distribución estadística, es decir, por la forma cómo ocurren en el tiempo. Si se supone que se producen con una tasa promedio constante y que son independientes una de otra, se dice que están distribuidas segun una ley de probabilidad de Poisson de parámetro λ en que λ es la tasa promedio de llegada por unidad de tiempo. Es decir se tiene que la probabilidad de que ocurran n llegadas en un tiempo T está dada por

$$P(n, T) = \frac{e^{-\lambda T} (\lambda T)^{n}}{n!}$$

La mayoría de las llegadas a sistemas reales pueden describirse mediante la distribución de Poisson. Sin embargo si la población total de unidades que pueden llegar es muy pequeña se incurre en errores debidos a la falta de independencia entre los eventos. Existen además otras distribuciones que pueden utilizarse entre ellas distribuciones empíricas o bien la distribución de Erlang, una de las más comunes pero que presenta una alta complejidad matemática.

Si las llegadas ocurren segun una distribucion de probabilidad de Poisson puede demostrarse que la variable aleatoria continua que mide el tiempo entre llegadas sucesivas tiene una distribución exponencial. Esto significa que la probabilidad de que el tiempo entre llegadas t sea menor o igual que un valor dado T se obtiene mediante

$$P(t \leq T) = 1 - e^{-\lambda T}$$

b) **La cola.** La cola que se forma en el sistema es de fundamental importancia para formular el modelo analítico. En efecto el modelo depende de si la cola tiene o no largo infinito de la forma de atención ("primero en llegar primero en ser atendido") y del comportamiento de las unidades que llegan a demandar el servicio. Por ejemplo si se hace una llamada telefónica y la central está ocupada, es necesario especificar si se desiste o se espera una linea.

El modelo analítico hace ciertas suposiciones respecto de las caracteristicas de la cola que es necesario tener en cuenta: el servicio es en orden de llegada, el largo de la cola es infinito y todas las llegadas esperan hasta ser atendidas.

Continua pag. 44

“quiero poder computacional para más de 100 escritorios, sin tener que comprar 100 microcomputadores”...

TOWER 32/800

El nuevo computador NCR de 32 bits hace posible que más de 100 usuarios interactivos compartan datos y aplicaciones, ahorrando tiempo y dinero.

El TOWER 32/800 de NCR trabaja tanto en una red de procesamiento distribuido en una organización con muchos usuarios, como en un sistema centralizado dedicado a una organización de tamaño medio.

El detalle de su potencial

■■■ Permite más de 100 usuarios interactivos compartiendo aplicaciones e información de oficina.

■■■ Arquitectura modular, con procesadores múltiples que le permite agregar mayor capacidad computacional cuando usted lo necesite.

■■■ Procesadores de especialización múltiple, optimizados para tareas de input/output, archivos y [illegible] de comunicaciones.

■■■ Compatible con otros equipos de la familia TOWER.

■■■ [illegible] de NCR [illegible]

■■■ [illegible] de NCR [illegible] de ventana [illegible] fácil uso.

■■■ [illegible] System [illegible] Ethernet, UNIX y [illegible] programas estándar.

■■■ Sistema de cinta y [illegible] de disco [illegible]

■■■ Acepta todos los lenguajes estándar.

■■■ Más de 3.500 programas [illegible] en el mercado.

■■■ Hasta 16 MB de memoria principal.



Poder Computacional

c) Los servidores La unidad de servicio debe especificarse mediante tres parámetros la distribución del tiempo de servicio, el numero de servidores y el número de fases

La distribución del tiempo de servicio generalmente puede suponerse exponencial usando la fórmula descrita para el tiempo entre llegadas de Poisson Sin embargo existen otras bastante frecuentes tales como un tiempo de servicio constante (caso de servidores automáticos) con distribución normal o con distribución uniforme

El sistema de espera puede tener uno o más servidores y justamente ése es uno de los parámetros que interesa estudiar a partir del modelo que presentamos midiendo el desempeño del sistema con diferente número de servidores Por otra parte si éstos se colocan en paralelo, se dice que el sistema tiene multiples servidores en una sola fase mientras que si se colocan en serie o en grupos serie-paralelo se dice que el sistema es de fases multiples

Las formulas para obtener las medidas de desempeño del sistema son las siguientes suponiendo que S es el numero de servidores E es el costo de espera y C es el costo de los servidores

$$\rho = \frac{\lambda}{s\mu}$$

$$Po = \frac{1}{\left[\sum_{n\ 0}^{s\ 1} (\lambda/\mu)^n/n!\right] + \frac{(\lambda/\mu)^s}{s!}(1-\lambda/s\mu)^{1}}$$

$$L_c = \frac{P_0(\lambda/\mu)^s\rho}{s!(1-\rho)^2}$$

$$Ls = L_c + \lambda/\mu$$
$$Wc = L_c/\lambda$$
$$Ws = W_c + 1/\mu$$
$$Ce = W_c * E$$
$$Cs = W_S * C$$
$$Ct = Ce + C_S$$

Formulación del modelo

El modelo analitico de un sistema de espera se define a traves de los siguientes parámetros y medidas de desempeno [2]

λ = tasa media de llegadas
μ = tasa media de servicio
ρ = factor de utilización del servidor (porcentaje del tiempo que los servidores están ocupados)

Pn = probabilidad de que n unidades se encuentren en el sistema en un instante cualquiera

Lc = largo promedio de la cola

Ls = número promedio de unidades en el sistema (esperando o recibiendo servicio)

Wc = tiempo medio de espera en la cola

Ws = tiempo medio dentro del sistema

Ca = costo de oportunidad del tiempo medio total

Cs = costo del servidor
Ct = costo total

Con estos parámetros se derivan la serie de fórmulas para obtener las medidas de desempeño sujetas a algunas restricciones impuestas para evitar la complejidad matemática Estas condiciones son las siguientes Los indicadores se calculan para un estado estable de largo plazo el estado estable implice que necesariamente la tasa de servicio es mayor que la de llegada existen multiples servidores en una sola fase las llegadas se distribuyen segun Poisson los tiempos de servicio son exponenciales y se atiende en estricto orden de llegada

Diseño del modelo en LOTUS 1-2-3

Las facilidades que presta el paquete LOTUS (u otras planillas electrónicas similares) para la automatización del modelo de colas hacen muy aconsejable su utilización En efecto la mayoria de las fórmulas pueden incorporarse directamente a una celda de la matriz entregando una precisión numérica adecuada y un formato de unidades de tiempo para los parámetros Wc y Ws Además es posible resolver un mismo problema simultáneamente para un buen numero de servidores facilitando el análisis mediante la selección automática de la solución que minimiza el costo total del sistema

Lotus facilita al analisia mediante la selacción automática de la solución óptima.

Sin embargo existen dos problemas que el LOTUS no puede rasolver directamente el cálculo del factorial (n!) y la sumatoria en el cálculo de Po (probabilidad de que no haya ninguna unidad en el sistema)

Para resolver esto se ha diseñado un mecanismo simple pero efectivo que se muestra a continuación

a) Factorial se construye una tabla auxiliar de dos columnas En la primera de ellas se coloca n y en la segunda n! tal como se muestra en la figura 1 Debe tomarse la precaución de incluir el 0! = 1 y 1! = 1 que son excepciones de la fórmula correspondiente que comienza a partir de la celda J11 y que simplemente corresponde a multiplicar el valor de la celda anterior en la misma columna

```
M1:                                                                  READY

       H       I        J         K        L          M          N
1                TABLAS AUXILIARES PARA CALCULO FACTORIAL Y SUMATORIA Po
2              ======================================================
3
4
5
6
7              n        n!                 G     SUMATORIA Po
8           ----------------           ----------------------
9              0        1                  1              1
10             1        1                  2            1.7
11             2        2                  3          1.945
12             3        6                  4   2.0021666667
13             4       24                  5   2.0121708333
14             5      120                  6   2.0133714167
15             6      720                  7   2.0137348181
16             7     5040                  8   2.0137511582
17             8    40320                  9    2.013752588
18             9   362880                 10   2.0137526992
19            10  3628800
20
06-Jan-86  01:50 PM                                       CAPS
```

figura 1

```
A12 [A6] 'MODELO DE COLAS DE ESPERA  M/M/S                           READY

    A    B        C          D           E          F          G
1   MODELO DE COLAS DE ESPERA  M/M/S
2   ==============================
3
4   DATOS DEL SISTEMA:
5   ---------------
6
7   TASA LLEGADA --------->          7 Vehiculos/dia
8   TASA SERVICIO -------->         10 Vehiculos/dia
9   COSTO SERVIDOR ------>        4500 $/dia
10  COSTO ESPERA -------->       56800 $/dia
11
12
13
14
15
16
17
18
19
20
06-Jan-86  01:45 PM                                       CAPS
```

figura 2

Continua en pág. 49





Llega a Chile

La Revolución Americana

El equipo ARC-TURBO, 100% Compatible,producido por American Research Corporation de California
DATAMERICA lo ofrece al más bajo precio del mercado: Configuración completa PC-Turbo a US$ 1 125 más IVA (*) y en las mejores condiciones de créditos a largo plazo

DATAMERICA

5 años en Chile

AV. PEDRO DE VALDIVIA 1642 - TELEFONOS 2516464 - 2516465 SANTIAGO.

*equivalente en moneda nacional.

por el valor de n en la misma fila.

Para obtener el factorial de un número bastará con usar la función @ VLOOKUP (x rango columna). Esta función recorre el rango de dos columnas usando x sobre la primera de ellas y extrae el valor asociado desde la columna especificada. Por ejemplo:

@ VLOOKUP (5,I9..J19,1) = 120

b) **Sumatoria**: se construye también una tabla auxiliar de dos columnas, como se aprecia en la figura 1. En la primera se coloca el valor de S (número de servidores) y en la segunda el valor correspondiente a la sumatoria de los términos hasta ese valor de S. Cuando S = 1, la celda M9 contiene el resultado de la fórmula

$$\frac{(\lambda/\mu)^n}{n!}$$

para n = 0. En la celda inferior (M10) está la suma del mismo término con n = 1 más el resultado de M9 y así sucesivamente. Con ello, bastará con usar la función @ VLOOKUP entrando con S al rango L9..M18 para obtener el valor total de la sumatoria correspondiente.

Resueltos estos problemas, la construcción del modelo es simple. La figura 2 muestra el área de entrada de datos (λ, μ, Ce, Cs), que está ubicada en las celdas D7..D10. La figura 3 muestra el área de cálculos, que contiene las fórmulas para obtener los indicadores de hasta 8 servidores del sistema. El rango B24..B31 está formateado como porcentaje (%) con un decimal para mostrar el porcentaje de utilización de los servidores, mientras que las demás columnas lo están con 8 valores decimales fijos en columnas de ancho 12. Las columnas de tiempos de espera usan el formato de hora español (D8), obtenido mediante el comando /RFDT3.

Dado que el «reloj» del formato D8 es de 24 horas, y el

A18: (W6) READY

	A	B	C	D	E	F	G
18							
19							
20							
21	NUM	UTIL					
22	SERV	SERV	Po	Lc	Ls	Wc	Ws
23							
24	1	70.0%	0.30000000	1.63333333	2.33333333	02:48:00	05:12:00
25	2	35.0%	0.48148148	0.09772080	0.79772080	00:10:03	02:34:03
26	3	23.3%	0.49515608	0.01123695	0.71123695	00:01:09	02:25:09
27	4	17.5%	0.49645212	0.00127699	0.70127699	00:00:08	02:24:08
28	5	14.0%	0.49657379	0.00013165	0.70013165	00:00:01	02:24:01
29	6	11.7%	0.49658437	0.00001213	0.70001213	00:00:00	02:24:00
30	7	10.0%	0.49658526	0.00000100	0.70000100	00:00:00	02:24:00
31	8	8.8%	0.49658530	0.00000007	0.70000007	00:00:00	02:24:00
32							
33							
34							
35							
36							
37							

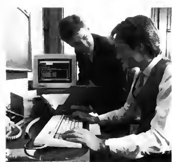
06-Jan-86 01:46 PM CAPS

figura 3

contenido de la celda puede variar entre 0 y 1 para obtener tiempos válidos. Las fórmulas de cálculo de los tiempos Wc y Ws deben ajustarse para obtener tiempos en ese rango de acuerdo al problema específico. En el caso del modelo que presentamos, el tiempo se divide por dos dado que la unidad de tiempo del problema usado como ejemplo es un turno de 12 horas.

La figura 4 muestra el área de cálculo de costos (A41..E54) y finalmente la figura 5 muestra el área de resultados del modelo. Para observar claramente el efecto de cambios en los parámetros del modelo conviene definir una ventana horizontal mediante /WWH y dejar el área de parámetros en la ventana superior y el área de resultados en la inferior, sin mostrar la matriz completa de 8 servidores.

El listado adjunto muestra el contenido de las celdas LOTUS que constituyen el modelo completo, para facilitar y revisar su digitación.



Un caso práctico

Supongamos el caso de un taller de reparaciones mecánicas en una faena minera de altura donde los camiones que transportan el mineral trabajan en un turno diario de doce horas y periódicamente sufren daños o fallas que obligan a su reparación inmediata. La tasa de llegada de vehículos al taller es de 7 camiones por día, mientras que el mecánico puede atender hasta diez vehículos por turno. Se conoce además el costo de oportunidad de tener los vehículos detenidos que corresponden al valor del mineral transportado por unidad de tiempo y que asciende a $ 56 800/día. El costo del mecánico es de $ 4 300/día.

Ingresando los datos al modelo analítico, se obtienen los resultados para un número variable de 1 a 8 mecánicos que se muestra en la figura 3. Por ejemplo, con un solo mecánico, éste está ocupado un 70% del tiempo, hay 1.63 vehículos como promedio esperando en la cola y 2.33 camiones en el sistema en un instante cualquiera. El tiempo medio de permanencia en la cola es de 2 horas 48 minutos, y en todo el sistema es de 5 horas 12 minutos.

Con dos mecánicos, en cambio, el tiempo en la cola se reduce a sólo 10 minutos, y el tiempo total a 2 horas y 34 minutos. Sin embargo, el porcentaje de utilización de los mecánicos baja a 35%. La matriz de análisis de costo de la figura 4 muestra claramente que el costo mínimo se encuentra con dos servidores, lo que se resume con todos los demás datos de la solución en el área de resultados de la figura 5.

Usando el modelo, es posible hacer un análisis de sensibilidad para responder las siguientes preguntas, entre otras: ¿qué sucede si el costo de oportunidad de tener los vehículos detenidos sube a $ 256 000 por día? Cambiando el valor correspondiente en el área de datos, se obtiene que no hay variación en la solución óptima, aunque sí hay un aumento del costo total. Por otra parte, ¿qué sucede si se contrata un mecánico más eficiente capaz de atender 14 vehículos en el día por el mismo costo? Reemplazando los valores en el modelo, se obtiene la nueva solución óptima: bastará con un solo mecánico, aunque el tiempo promedio en la cola aumente de 10 a 56 minutos.

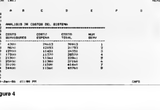


figura 4

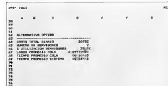


figura 5



Conclusiones

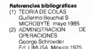
Sin duda, el hecho de modelar el sistema de espera en una planilla electrónica facilita mucho el análisis, y coloca una herramienta analítica de gran valor al alcance de usuarios que conocen este tipo de software.

pero no los lenguajes tradicionales de programación. Por otra parte, es necesario siempre tener en cuenta que las simplificaciones matemáticas muchas veces hacen inaplicable un modelo a situaciones reales, lo que quedará a juicio de los usuarios de esta herramienta.

Referencias bibliográficas

(1) TEORIA DE COLAS
Guillermo Beuchat S.
MICROBYTE, mayo 1985

(2) ADMINISTRACION DE OPERACIONES
George Schroeder
Ed. LIMUSA, México 1975



Guillermo Beuchat S. es Ingeniero Civil Industrial de la U. de Chile, habiéndose especializado en el área de Informática y Sistemas. Su principal interés está en el uso de la computación como herramienta estratégica en la gestión de empresas y en la administración eficiente de los recursos de información. Se ha desempeñado como consultor independiente en microcomputadores, consultor de Auditoría Computacional y relator de seminarios de microcomputación para ejecutivos de Price Waterhouse y profesor auxiliar de computación en la Universidad Gabriela Mistral. Actualmente trabaja como Analista en Shell Chile S.A.C.I. y es miembro del comité editorial de MICROBYTE.


```
A1: [W6] 'MODELO DE COLAS DE ESPERA  M/M/S
I1: '       TABLAS AUXILIARES PARA CALCULO
    FACTORIAL Y SUMATORIA Po
A2: [W6] \=
B2: [W6] \=
C2: [W12] \=
D2: [W12] \=
I2: \=
J2: [W12] \=
K2: \=
L2: \=
M2: [W14] \=
N2: \=
A4: [W6] 'DATOS DEL SISTEMA:
A5: [W6] \-
B5: [W6] \-
C5: [W12] '-----
A7: [W6] 'TASA LLEGADA --------->
D7: [W12] 7
E7: [W12] 'Vehiculos/dia
I7: ^n
J7: [W12] ^n'
L7: ^n
M7: [W14] 'SUMATORIA Po
A8: [W6] 'TASA SERVICIO -------->
D8: [W12] 10
E8: [W12] 'Vehiculos/dia
I8: \-
J8: [W12] \-
L8: \-
M8: [W14] \-
A9: [W6] 'COSTO SERVIDOR ------->
D9: [W12] 4300
E9: [W12] '$/dia
I9: 0
J9: [W12] 1
L9: 1
M9: [W14] (($D$7/$D$8)^(L9-1))/@VLOOKUP
    (L9-1,$I$9..$J$19,1)
A10: [W6] 'CDSTO ESPERA --------->
B10: [W12] 56800
E10: [W12] '$/dia
I10: 1
J10: [W12] 1
L10: 2
M10: [W14] (($D$7/$D$8)^(L10-1))/@VLOOKUP
    (L10-1,$I$9..$J$19,1)+M9
A11: [W6] '
I11: 2
J11: [W12] +J10*I11
L11: 3
M11: [W14] (($D$7/$D$8)^(L11-1))/@VLOOKUP
    (L11-1,$I$9..$J$19,1)+M10
I12: 3
J12: [W12] +J11*I12
L12: 4
M12: [W14] (($D$7/$D$8)^(L12-1))/@VLOOKUP
    (L12-1,$I$9..$J$19,1)+M11
I13: 4
J13: [W12] +J12*I13
L13: 5
M13: [W14] (($D$7/$D$8)^(L13-1))/@VLOOKUP
    (L13-1,$I$9..$J$19,1)+M12
I14: 5
J14: [W12] +J13*I14
L14: 6
M14: [W14] (($D$7/$D$8)^(L14-1))/@VLOOKUP
    (L14-1,$I$9..$J$19,1)+M13
I15: 6
J15: [W12] +J14*I15
L15: 7
M15: [W14] (($D$7/$D$8)^(L15-1))/@VLOOKUP
    (L15-1,$I$9..$J$19,1)+M14
I16: 7
J16: [W12] +J15*I16
L16: 8
M16: [W14] (($D$7/$D$8)^(L16-1))/@VLOOKUP
    (L16-1,$I$9..$J$19,1)+M15
I17: 8
J17: [W12] +J16*I17
L17: 9
M17: [W14] (($D$7/$D$8)^(L17-1))/@VLOOKUP
    (L17-1,$I$9..$J$19,1)+M16
I18: 9
J18: [W12] +J17*I18
L18: 10
M18: [W14] (($D$7/$D$8)^(L18-1))/@VLOOKUP
    (L18-1,$I$9..$J$19,1)+M17
I19: 10
J19: [W12] +J18*I19
A21: [W6] ^NUM
B21: [W6] ^UTIL
A22: [W6] ^SERV
B22: [W6] ^SERV
C22: [W12] ^Po
D22: [W12] ^Lc
E22: [W12] ^Ls
F22: [W12] ^Wc
G22: [W12] ^Ws
A23: [W6] \-
B23: [W6] \-
C23: [W12] \-
D23: [W12] \-
E23: [W12] \-
F23: [W12] \-
G23: [W12] \-
A24: [W6] 1
B24: (P1) [W6] +$D$7/(A24*$D$8)
C24: (F0) [W12] 1/((@VLOOKUP(A24,$L$9..
```

```
         $M$18,1))+(($D$7/$D$8)^A24)/
         (@VLOOKUP(A24,$I$9..$J$19,1))*
         (1-$D$7/$D$8/A24)))
D24:  (F8) [W12] (C24*($D$7/$D$8)^A24*B24)
         /(@VLOOKUP(A24,$I$9..$J$19,1)*(1-B24)^2)
E24:  (F8) [W12] +D24+$D$7/$D$8
F24:  (D8) [W12] (D24/$D$7)/2
G24:  (D8) [W12] (F24*2+2/$D$8)/2
A25:  [W6] 2
B25:  (P1) [W6] +$D$7/(A25*$D$8)
C25:  (F8) [W12] 1/((@VLOOKUP(A25,$L$9..
         $M$18,1))+(($D$7/$D$8)^A25)/(@VLOOKUP(
         25,$I$9..$J$19,1)*(1-$D$7/$D$8/A25)))
D25:  (F8) [W12] (C25*($D$7/$D$8)^A25*B25)
         /(@VLOOKUP(A25,$I$9..$J$19,1)*(1-B25)^2)
E25:  (F8) [W12] +D25+$D$7/$D$8
F25:  (D8) [W12] (D25/$D$7)/2
G25:  (D8) [W12] (F25*2+2/$D$8)/2
A26:  [W6] 3
B26:  (P1) [W6] +$D$7/(A26*$D$8)
C26:  (F8) [W12] 1/((@VLOOKUP(A26,$L$9..
         $M$18,1))+(($D$7/$D$8)^A26)/
         (@VLOOKUP(A26,$I$9..$J$19,1))*
         (1-$D$7/$D$8/A26)))
D26:  (F8) [W12] (C26*($D$7/$D$8)^A26*B26)
         /(@VLOOKUP(A26,$I$9..$J$19,1)*
         (1-B26)^2)
E26:  (F8) [W12] +D26+$D$7/$D$8
F26:  (D8) [W12] (D26/$D$7)/2
G26:  (D8) [W12] (F26*2+2/$D$8)/2
A27:  [W6] 4
B27:  (P1) [W6] +$D$7/(A27*$D$8)
C27:  (F8) [W12] 1/((@VLOOKUP(A27,$L$9..
         $M$18,1))+(($D$7/$D$8)^A27)/
         (@VLOOKUP(A27,$I$9..$J$19,1))*
         (1-$D$7/$D$8/A27)))
D27:  (F8) [W12] (C27*($D$7/$D$8)^A27*
         B27)/(@VLOOKUP(A27,$I$9..
         $J$19,1)*(1-B27)^2)
E27:  (F8) [W12] +D27+$D$7/$D$8
F27:  (D8) [W12] (D27/$D$7)/2
G27:  (D8) [W12] (F27*2+2/$D$8)/2
A28:  [W6] 5
B28:  (P1) [W6] +$D$7/(A28*$D$8)
C28:  (F8) [W12] 1/((@VLOOKUP(A28,$L$9..
         $M$18,1))+(($D$7/$D$8)^A28)/
         (@VLOOKUP(A28,$I$9..$J$19,1))*
         (1-$D$7/$D$8/A28)))
D28:  (F8) [W12] (C28*($D$7/$D$8)^A28*D28)
         /(@VLOOKUP(A28,$I$9..$J$19,1)*
         (1-B28)^2)
E28:  (F8) [W12] +D28+$D$7/$D$8
F28:  (D8) [W12] (D28/$D$7)/2
G28:  (D8) [W12] (F28*2+2/$D$8)/2
A29:  [W6] 6
B29:  (P1) [W6] +$D$7/(A29*$D$8)
C29:  (F8) [W12] 1/((@VLOOKUP(A29,$L$9..
         $M$18,1))+(($D$7/$D$8)^A29)/
         (@VLOOKUP(A29,$I$9..$J$19,1))*
         (1-$D$7/$D$8/A29)))
D29:  (F8) [W12] (C29*($D$7/$D$8)^A29*B29)
         /(@VLOOKUP(A29,$I$9..$J$19,1)*
         (1-B29)^2)
E29:  (F8) [W12] +D29+$D$7/$D$8
F29:  (D8) [W12] (D29/$D$7)/2
G29:  (D8) [W12] (F29*2+2/$D$8)/2
A30:  [W6] 7
B30:  (P1) [W6] +$D$7/(A30*$D$8)
C30:  (F8) [W12] 1/((@VLOOKUP(A30,$L$9..
         $M$18,1))+(($D$7/$D$8)^A30)/
         (@VLOOKUP(A30,$I$9..$J$19,1))*
         (1-$D$7/$D$8/A30)))
D30:  (F8) [W12] (C30*($D$7/$D$8)^A30*B30)
         /(@VLOOKUP(A30,$I$9..$J$19,1)*
         (1-B30)^2)
E30:  (F8) [W12] +D30+$D$7/$D$8
F30:  (D8) [W12] (D30/$D$7)/2
G30:  (D8) [W12] (F30*2+2/$D$8)/2
A31:  [W6] 8
B31:  (P1) [W6] +$D$7/(A31*$D$8)
C31:  (F8) [W12] 1/((@VLOOKUP(A31,$L$9..
         $M$18,1))+(($D$7/$D$8)^A31)/
         (@VLOOKUP(A31,$I$9..$J$19,1))*
         (1-$D$7/$D$8/A31)))
D31:  (F8) [W12] (C31*($D$7/$D$8)^A31*B31)
         /(@VLOOKUP(A31,$I$9..$J$19,1)*
         (1-B31)^2)
E31:  (F8) [W12] +D31+$D$7/$D$8
F31:  (D8) [W12] (D31/$D$7)/2
G31:  (D8) [W12] (F31*2+2/$D$8)/2
A41:  [W6] 'ANALISIS DE COSTOS DEL SISTEMA
A42:  [W6] \=
B42:  [W6] \=
C42:  [W12] \=
D42:  [W12] \=
E42:  [W12] \=
A44:  [W6]  COSTO
C44:  [W12] ^COSTO
D44:  [W12] ^COSTO
E44:  [W12] ^NUM
A45:  [W6] ^SERVIDORES
C45:  [W12] ^ESPERA
D45:  [W12] ^TOTAL
E45:  [W12]  SERV
A46:  [W6] \-
B46:  [W6] \-
C46:  [W12] \-
D46:  [W12] \-
E46:  [W12] \-
A47:  (F0) [W6] +E47*$D$9
C47:  (F0) [W12] +G24*2*$D$10
D47:  (G) [W12] @INT(A47+C47+0.5)
E47:  (G) [W12] 1
A48:  (F0) [W6] +E48*$D$9
C48:  (F0) [W12] +G25*2*$D$10
D48:  (G) [W12] @INT(A48+C48+0.5)
E48:  (G) [W12] 2
A49:  (F0) [W6] +E49*$D$9
C49:  (F0) [W12] +G26*2*$D$10
D49:  (G) [W12] @INT(A49+C49+0.5)
E49:  (G) [W12] 3
A50:  (F0) [W6] +E50*$D$9
C50:  (F0) [W12] +G27*2*$D$10
D50:  (G) [W12] @INT(A50+C50+0.5)
E50:  (G) [W12] 4
A51:  (F0) [W6] +E51*$D$9
C51:  (F0) [W12] +G28*2*$D$10
D51:  (G) [W12] @INT(A51+C51+0.5)
E51:  (G) [W12] 5
A52:  (F0) [W6] +E52*$D$9
C52:  (F0) [W12] +G29*2*$D$10
D52:  (G) [W12] @INT(A52+C52+0.5)
E52:  (G) [W12] 6
A53:  (F0) [W6] +E53*$D$9
C53:  (F0) [W12] +G30*2*$D$10
D53:  (G) [W12] @INT(A53+C53+0.5)
E53:  (G) [W12] 7
A54:  (F0) [W6] +E54*$D$9
C54:  (F0) [W12] +G31*2*$D$10
D54:  (G) [W12] @INT(A54+C54+0.5)
E54:  (G) [W12] 8
A62:  [W6] 'ALTERNATIVA OPTIMA
A63:  [W6] \-
B63:  [W6] \-
C63:  [W12] \-
D63:  [W12] \-
A64:  [W6] 'COSTO TOTAL DIARIO
D64:  (F0) [W12] @MIN(D47..D54)
A65:  [W6] 'NUMERO DE SERVIDORES
D65:  (F0) [W12] @IF(D47=D64,1,@IF(D48=
         D64,2,@IF(D49=D64,3,@IF(D50=D64,4,
         1=D64,5,@IF(D52=D64,6,@IF
         (D53=D64,7,8)))))))
A66:  [W6] '% UTILIZACION SERVIDORES
D66:  (P1) [W12] @VLOOKUP($D$65,$A$24..
         $B$31,1)
A67:  [W6] 'LARGO PROMEDIO COLA
D67:  (F8) [W12] @VLOOKUP($D$65,$A$24..
         $G$31,3)
A68:  [W6] 'TIEMPO PROMEDIO COLA
D68:  (D8) [W12] @VLOOKUP($D$65,$A$24..
         $G$31,5)
A69:  [W6] 'TIEMPO PROMEDIO SISTEMA
D69:  (D8) [W12] @VLOOKUP($D$65,$A$24..
         $G$31,6)
```

ELCA

Sistema Uniwell S-15

INSTITUCIONES

DISTRIBUIDORAS SUPERMERCADOS

FARMACIAS FERRETERIAS

TIENDAS DE DEPARTAMENTOS

ELCA presenta su nueva línea de Sistemas UNIWELL con la más moderna tecnología computacional aplicada al control de transacciones en el comercio mayorista y detallista, para cada tipo de negocio o actividad.

Cajas Registradoras de vanguardia, capaces de actuar como unidades independientes o trabajar como un Sistema en Línea, conectado directamente a su computador en Tiempo Real.

Los Sistemas ELCA UNIWELL han sido diseñados pensando en el crecimiento de su negocio. Lectores de Código de Barras, Scanner de Mesón, Visor Alfanumérico, Tarjetas de Crédito, Impresoras de Documentos, etc.

Gane eficiencia y control con los nuevos Sistemas ELCA UNIWELL, capaces de informar minuto a minuto sobre el estado de sus stocks en bodega y en estanterías, reportes de ventas, costos y reposición de mercaderías, control de créditos, preferencias y hábitos de clientes, etc., en el momento que Ud. los requiera.

Decídase hoy a incorporar esta moderna tecnología, comience con un económico sistema S-15 básico (capaz de controlar miles de ítems), para luego crecer a una solución integral computarizada, a un costo muy inferior al que Ud. imagina.

